Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels
UNIVERSITAT POLITÈCNICA DE CATALUNYA

MASTER THESIS

TITLE: Analysis of Perturbation incidence in the calculation of trajectories in Ephemeris Model

MASTER DEGREE: Master's degree in Aerospace Science and Technology

AUTHOR: Omar Diab Pascual

DIRECTOR: Ricard González Cinca

DATE: November, 8th 2019

Title: Analysis of Perturbation incidence in the calculation of trajectories in Ephemeris Model

Author: Omar Diab Pascual

Advisor: Ricard González Cinca

Date: November, 8th 2019

Overview

The main goal of this thesis is to deepen the understanding of the influence that the various perturbations exert on the trajectory calculations performed by the JPL ephemeris model to provide high-fidelity ephemeris in support of spacecraft navigation and other activities related to Solar System bodies. One of the drawbacks behind such a complex method are the large machine-oriented technical requirements it has in terms of time and computational cost. The algorithm that integrates the full dynamical model of the N-Body differential equations of motion requires a continuous update of the dynamical states (positions and velocities) of a large number of bodies for each new integration step, regardless of the perturbation model applied or the nature and diversity of the sources of perturbations other than those gravitational considered.

This work focuses especially to the gravitational implications of interplanetary trajectories, based on the fact that within its main and longest phase (the cruise phase) a spacecraft is essentially affected by a multi-body attraction. A procedure has been conducted to build a tool that, applied to any given interplanetary trajectory, has the ability of predicting which gravitational perturbations are not relevant and can thus be neglected without a significant loss of accuracy. Validation tests based on the BepiColombo interplanetary ephemeris trajectory planned mission proved that no significant differences are found when comparing the discrepancies exhibited by a numerical propagation of its initial conditions with the N-Body differential equations of motion, including all planetary perturbations or only those indicated as relevant by the developed 2D maps.

A comparison is also carried out between two of the most popular special perturbation techniques, Cowell and Encke, which demonstrated that round off errors take its toll even working in double precision and showed that Encke's method performs much better for interplanetary trajectories where there is a large difference in forces between the central dominant body and the perturbing bodies.

Finally, a simple debugging test carried out by *MatLab* showed that both time and computational resources (amount of data) are sensibly reduced by not considering those negligible bodies suggested by the application of maps and tests. Thus, this result offer the possibility to save time both in the access and in the download of ephemeris data from the platform while avoiding loading times to create large data matrices in the workspace.

Título: Análisis de la incidencia de las Perturbaciones en el cálculo de trayectorias en el Modelo de Efemérides

Autor: Omar Diab Pascual

Director: Ricard González Cinca

Fecha: 8 de Noviembre de 2019

Resumen

El objetivo principal de esta tesis es profundizar en la comprensión de la influencia que las diversas perturbaciones ejercen en los cálculos de trayectoria realizados por el modelo de efemérides JPL que proporciona efemérides de alta fidelidad en apoyo de la navegación de naves espaciales y otras actividades relacionadas con los cuerpos del Sistema Solar. Una de las desventajas de un método tan complejo son los grandes requisitos técnicos orientados a la computación que tiene en términos de tiempo y costes de cálculo. El algoritmo que integra el modelo dinámico completo de las ecuaciones diferenciales de movimiento de N cuerpos requiere una actualización continua de los estados dinámicos (posiciones y velocidades) de un gran número de cuerpos celestiales para cada nuevo paso de integración, independientemente del modelo de perturbación aplicado o de la naturaleza y diversidad de las fuentes de perturbaciones distintas de las gravitacionales consideradas.

Este trabajo se centra especialmente en las implicaciones gravitatorias de las trayectorias interplanetarias, basadas en el hecho de que dentro de su fase principal y más larga (la fase de crucero) una nave espacial se ve esencialmente afectada por una atracción multicuerpo. Se ha llevado a cabo un procedimiento para construir una herramienta que, aplicada a cualquier trayectoria interplanetaria, tiene la capacidad de predecir qué perturbaciones gravitatorias no son relevantes y, por lo tanto, pueden ser ignoradas sin una pérdida significativa de precisión.

Las pruebas de validación basadas en la misión planificada de trayectoria de efemérides interplanetarias de BepiColombo demostraron que no se encuentran diferencias significativas al comparar las discrepancias exhibidas por una propagación numérica de sus condiciones iniciales con las ecuaciones diferenciales de movimiento incluyendo todas las perturbaciones planetarias o sólo aquellas indicadas como relevantes por los mapas 2D desarrollados.

También se realiza una comparación entre dos de las técnicas de perturbación especiales más populares, Cowell y Encke, que demostraron que los errores de redondeo se cobran su peaje incluso trabajando con doble precisión y que el método de Encke funciona mucho mejor para trayectorias interplanetarias en las que existe una gran diferencia de fuerzas entre el cuerpo central dominante y los cuerpos perturbadores.

Finalmente, una sencilla prueba de rendimiento del código, llevada a cabo por MatLab, mostró que tanto el tiempo como los recursos computacionales (cantidad de datos) se reducen sensiblemente al no considerar aquellos cuerpos previamente sugeridos como poco significativos por la aplicación de mapas y pruebas. De esta manera, este resultado ofrece la posibilidad de ahorrar tiempo tanto en el acceso como en la descarga de datos de efemérides desde la plataforma, evitando tiempos de carga para crear grandes matrices de datos en el espacio de trabajo.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AU	Astronomical Units
AD	Anno Domini
2BP	Two – Body Problem
BC	Before Christ
BOL	Beginning of Life
CM	Centre of Mass
CT	Central Time
CRTBP	Circular Restricted Three – Body Problem
EPS	Electric Propulsion System
ESA	European Space Agency
GEO	Geocentric Transfer Orbit
GPS	Global Positioning System
IAU	International Astronomical Union
IRF	International Reference Frame
JPL	Jet Propulsion Laboratory
MFR	Mass Factor Ratio
NASA	National Aeronautics and Space Administration
NBP	N – Body Problem
PZ	Perturbation Zones
PS	Perturbation Sphere
RTBP	Restricted Three – Body Problem
SC	Spacecraft
SOI	Sphere of Influence
SSB	Solar System Barycentre

REPORT OUTLINE

The work is organised in a total of 8 chapters, where each of them – except the first and the last one containing the introduction and the conclusions respectively – is presented in such a way that the necessary theoretical basis are established for the followings ones, or simply an attempt is made to follow a logical chronology of the developments and analyses carried out to achieve the main objective of this study.

Chapter 1 ➔ The opening chapter serves to present the motivations that stimulated this work and the statement of the basic objectives pursued, seasoned with a small historical background that offers a pleasant introduction to Astrodynamics.

Chapter 2 ➔ This chapter emanates a strong theoretical and mathematical character. The reason why it will be carefully developed rather than just introduce the necessary equations – with the corresponding citations – lies in the fact that it has been considered crucial to settle the foundations of the complex dynamic nature of the NBP equations of motion, as well as the physical meanings and implications of each term, in order to follow all further developments and analyses. Understanding aspects such as the choice of a suitable reference system is capital, since the ephemeris data gathered from the HORIZONS platform will refer to a reliable coordinate system not always used. In addition, it is very important to get a proper idea of what a perturbation is, what types exist or how to introduce them into the equations of motion.

Chapter 3 ➔ This chapter presents two of the most commonly used special perturbation models – Cowell and Encke. They form the basis of the most accurate machine-generated planetary ephemeris. Latter, they will be implemented in MatLab to simulate a real interplanetary ephemeris trajectory that validates the results provided by the procedure developed in this thesis.

Chapter 4 ➔ Once all the necessary technical issues are on the table, this chapter will present the strategy envisaged to carry out the steps leading to the acquisition of the information required for subsequent analyses. This “roadmap” will be summarised in a schematic flowchart.

Chapter 5 ➔ It will take the first step necessary to apply the tool to be developed. Once the concept of perturbation acceleration threshold has been introduced – which is the upper limit magnitude of the vector sum of all perturbing accelerations that a propelled object can admit without significantly affecting its motion – a realistic estimate will be made to represent a modern spacecraft. After two iterations, BepiColombo is selected as the reference.

Chapter 6 ➔ Here, maps representing the volume of control under study of the solar system will be obtained and analysed, showing the areas in which a spacecraft, accelerated with respect to the Sun, would be subjected to planetary perturbations above its established acceleration threshold. This will subsequently lead to the development of a criterion for the identification of the relevant disturbances in a given interplanetary trajectory.

Chapter 7 ➔ Once the tool/procedure is ready, the next natural step is to proceed with its validation. This will be done by simulating a real interplanetary ephemeris trajectory through the numerical integration of the NBP equations of motion, using the two special perturbation models presented in Chapter 3. The purpose of the process is to demonstrate that no significant difference is found in the motion of the spacecraft by propagating the initial conditions of the ephemeris when considering all the planets as perturbations or simply those selected ones.

Chapter 8 ➔ The ending chapter will reflect an overall conclusion, based on the analyses conducted, and will propose future lines of work that may be of interest, either to improve the results and performance of the tool developed, or to complete it and make it more robust.

CHAPTER 1. INTRODUCTION

This first chapter deals with building the foundations of the present work, beginning with a brief historical description of what concerns the field of Astrodynamics. Then, the topic and background that motivated this study will be introduced and the objectives proposed for this thesis will be presented.

1.1. A little bit of history: The origins of Astrodynamics

The word "*Astronomy*" etymologically means "*the law of the stars*" and the word "*planet*" originally comes from the Greek word "*wanderer*". That was because, at first glance, the planets seem to wander figuratively among the stars. This was one of the earliest astronomical observations that, indeed, was not well understood at first.

Still within the ancient Greek era, Aristarchus of Samos was, as early as the second century BC, the first to realise that the planets must revolve around the Sun, assuming it must be the dominant central body of the solar system [1]. Nevertheless, the current of opinion turned towards the adoption of the historically conflictive model centred on the Earth, until Copernicus (1473–1543) finally rediscovered the heliocentric system – already in the mid-16th century – and published his work under the title of *De Revolutionibus orbium coelestium*. His compiled tables, which contain a description of the planetary motion, remained useful until a much more accurate measurements performed by the later great astronomer Tycho Brahe (1546-1601) superseded them. Following the summary of the chronology of events, as the reader probably is aware of, such was the dark atmosphere of that ages that, even the decisive observations of Galileo Galilei (1564-1642) in 1610, failed to change the Church's position. Finally, in 1687 Sir Isaac Newton (1642-1727) published his widely known *Philosophiae Naturalis Principia Mathematica*, which provided a solid mathematical foundation for describing the planetary motion in terms of geometric elements and laid to rest forever the Earth – centred concept of the solar system. In fact, such a gravitational model gave evidence to the three laws of the planetary motion that Johannes Kepler (1571-1630) had formulated almost a century earlier. A curious fact, always related to Newton's brilliant work, was the need to invent his own version of what is nowadays a branch of mathematics known as differential calculus, which was completely necessary to deal with the complicated formulation that governs the nature of motion. However, the more compact methods developed by Gottfried Leibniz (1646-1716) soon replaced Newton's tool for addressing dynamic problems. The unification of Celestial Mechanics with Gravitational Theory was finally completed by Pierre – Simon de Laplace (1749-1827) who applied the differential calculus to the motion of celestial bodies in his famous five-volume work *Traité de Mécanique Céleste*.

The aforementioned milestone gave way to the study of the motion of an artificial body within the celestial gravitational field, which is currently being studied in depth by a science-derived branch called Astrodynamics. The first

and simplest dynamical model emerged to describe the motion of such artificial conceptual body (e.g. a spacecraft or a satellite) barely accounts for one attractor at a time. This model – which its complete solution it is not, unless it is assumed that the two bodies involved are regular – is known as the *two–body problem*[1] (hereinafter simply referred as 2BP). It has been used reliably since the 17th century and continues to be used for the first rough calculations in the preliminary design of space missions. The next level of fidelity, in terms of describing natural motion, lies in the dynamical model corresponding to the restricted three-body problem (RTBP), which deals with two main attractors (called primaries) and the mentioned non-massive artificial object. This formulation was first initiated by Leonhard Euler (1707-1783) in relation to his lunar theories, and continued by Joseph-Louis Lagrange (1736-1813) and still has no closed–form solution, despite the great efforts made by the entire scientific community. Nevertheless, important contributions should be mentioned, as they facilitated many of the subsequent analyses. Among them, Euler introduced the synodic rotating reference frame instead of using an inertial one, thus greatly simplifying the complex equations of motion. In addition, Lagrange discovered the existence of five equilibrium points[2]; Carl Gustav Jacob Jacobi (1804-1851) found an integral of motion[3], which is the only magnitude conserved in the particular case of RTBP called the circular restricted three-body problem (CRTBP) and George William Hill (1838-1914) mapped the region of the associated coherent motion. Ultimately, Jules Henri Poincaré (1854-1912) made an extraordinary contribution with his *Le Méthodes Nouvelles de la Mécanique Céleste*, which laid the basis of *Dynamical System Theory*[4], a mathematical area dedicated to describing the behaviour of complex non-linear differential equation systems.

Today, extensive and active research efforts are focused on deepening, analysing, and accurately solving the models mentioned above which, indeed, simplify the real N-body problem that nature unconsciously applies by itself, extending the influence on the physics of the movement of an artificial object in space to – as a minimum – all the bodies present in the solar system.

Before ending this section, it is worth mentioning a last but very relevant contribution made by Albert Einstein (1879-1955) with his *General Theory of Relativity*, which states that all matter curves space and without which, some anomalies such as that observed in the motion of Mercury in its perihelion could not be explained.

---

[1] Also commonly known as *Kepler problem.*

[2] Usually referred to as the own name of its discoverer. Sometimes they can be also found as libration points or simply as fixed or stationary points.

[3] It can also be found as Jacobi's constant

[4] It is widely applied to CRTBP and the four – body problem (CR4BP) to get important solutions for real problems (e.g, Periodic orbits or Invariant manifolds, among others). It will no be used in the present work. Nevertheless, if the reader is interested in deepening the concepts, two good bibliography sources can be found in the books *Introduction to Applied Nonlinear Systems and Chaos* [B.1] and *Nonlinear Differential Equations and Dynamical Systems* [B.2].

1.2. Motivations and background

In the previous historical review it was stated that the formulation of the N-body problem describes the true natural motion of a mass particle in a vast and complex multi-body gravitational field created by the rest of the N-1 bodies. Such a statement is based on Newton's proven Law of Universal Gravitation. A development will be made to obtain these equations in the next chapter, for the sake to understand the appearance of its different terms, physical meanings and implications, which is crucial to adequately follow all further developments and analyses. For the time being, suffice it to say that, achieving a complete understanding of what is happening in the nature of motion is more a desire still today than a reality (even ignoring some perturbations). Having been the subject of thorough studies in recent decades, the differential equations of motion do not formally have a closed-form solution yet – except in really simplified cases – which stimulated many other studies that were carried out numerically. Focussing on the problem from the latter perspective takes its toll on losing the generality and depth that analytic formulations usually offer. As is often the case in the world of science, the most manageable and fruitful models that describe reality are "*too good to be true*" while, on the contrary, those who rigorously consider every single detail are, unfortunately, "*too true to be good*"…

On the other hand, another problem inherent to the general formulation arises from the strong nonlinear character exhibited by the system[5] of differential equations. This difficulty is based on the fact that, even if a small arbitrary perturbation is introduced in the initial conditions, the numerical integration of the equations of motion – generally known as trajectory propagation – could lead to large deviations from the expected solution (which is a characteristic of chaotic dynamical systems).

As a result of all the complexity mentioned above, deep learning has been achieved; first, using the most basic version of the problem (2BP) to build later more complicated ones step-by-step. This model – which is nothing more than the NBP equations of motion simplified to the maximum – has been extensively used in the past and is still used to compute a first approximation of a trajectory. In addition, in the early stages of feasibility studies in space mission design, interplanetary trajectories are constructed by patching together the different legs, where other gravitational attractions apart from that of the dominant body[6] are no longer relevant and can therefore be neglected. In any case, the trajectories designed by applying this model will never be able to offer a fair degree of accuracy or fully exploit the potential of the gravitational dynamics that encloses the general formulation.

[5] In a later development it will be observed that, the NBP equations of motion, actually come from a single vector expression that can be broken down into its Cartesian – or Polar – components, which confers them such a state of "*system"*.

[6] It is based on a concept called Sphere of influence. The lecture of Kemble, S. [B.3] is recommended for proper understanding.

When the NBP motion equations are stated, it will be quickly noted that their second term corresponds to the perturbing effects exerted by the N-1 bodies on a mass particle accelerated relative to the dominant body (commonly the Sun). It is important to bear in mind that perturbations come not only from gravitational sources, but also from the propulsion of the artificial body itself (typically a spacecraft in the case of interplanetary trajectories) or from other external forces[7] such as the produced by the non-sphericity of the masses, the solar radiation pressure, atmospheric drag, magnetic fields or even relativistic effects. Considering a single perturbing body, a much better description of reality than that provided by 2BP is achieved through the RTBP model. If the mass of the artificial body is neglected in front of the other two (the primaries) and a rotating frame is chosen – centred in the barycentre of the straight line that joins the centres of mass of the primaries – then the CRTBP[8] emerges. Being this model much more accurate, despite coming barely from the addition of a third body, it has been widely studied, analysed and applied in many cases to get a variety of useful solutions such as the quasi–stable motion around libration points, generation of invariant manifolds or ballistic capture, inter alia, which led to the development of robust and precise low–energy transfer manoeuvres.

At this point, the subject matter of this work begins to be closely related to what constitutes a next step in the modelling trend: the JPL ephemeris model[9]. It corresponds, as Parker and Anderson affirm in their document for the design of a low-energy lunar trajectory transfer [3], to the most accurate model of the Solar System used in their work. Also, in Newhall's words [4] the model provides high-precision numerically integrated planetary and lunar ephemeris[10] in support of spacecraft navigation and other activities related to solar system bodies.

The aim of this thesis is to deepen the understanding of the influence that the various perturbations exert in the trajectory calculations made through this powerful model, placing a particular emphasis on the interplanetary trajectories carried out by modern spacecraft. As the HORIZONS[11] platform expressly indicates, the interplanetary trajectories provided by navigation teams reflect the full dynamical model, including thruster firings, solar pressure, extended spherical harmonic gravity fields, atmospheric drag, and whatever other dynamic model is used for navigation [5]. Making use of the high-fidelity framework offered by this tool, a vast number of references can be found in the literature, providing optimised interplanetary trajectories for a quite wide range of selections. It is worth mentioning the thesis of Diogene, [6] where a numerical procedure is developed to automatically refine trajectories, previously designed

[7] They will be introduced and briefly described in *Chapter 2*, *Section 2.2*, *Sub-section 2.2.2. External forces. Main sources of perturbation*.

[8] Not being object of study in this work, the reader can easily find a great quantity of papers, lectures and books, where this model is carefully developed step by step. For instance, a recommended source in literature is the thesis of Grebow [2].

[9] It will be presented and described appropriately in the next chapter, Section 2.4.

[10] The current update includes also comets, asteroids, natural satellites, spacecraft trajectories and several dynamical points such as the Earth-Sun L_1, L_2, L_4, L_5 and systems barycentre's.

[11] This is the name of the On-line Ephemeris System tool, created by the JPL of the California Institute of Technology, from where the different ephemeris can be acquired.

in RTBP, using the ephemeris model. In Gómez et al. [7] efficient methods are used to continue the refinement of a semi – analytical HALO orbit in high-fidelity solar system models. Two different articles (Tang et al. [8] and Lian et al. [9]) deepened in the analysis of the dynamics around the Lagrange collinear points of the Earth-Moon system in the dynamical panorama of the complete solar system. A work that exhibits a high degree of similarity with the basis underlying this thesis can be found in [10], which determines the effectiveness of the most used special perturbation methods in computing high eccentric hyperbolic orbit trajectories of comets and real artificial satellites. What differentiates this work is that, such special techniques, will be applied only in the last part to test the validity and limitations of a tool/procedure that will allow detecting, for a high-fidelity ephemeris interplanetary trajectory, which perturbations are not relevant and, therefore, can be neglected without a significant loss of accuracy.

Having qualitatively established the depth of the JPL ephemeris model with respect to the number of perturbation sources it has integrated to describe the behaviour of a high-fidelity system, one can easily guess how large the requirements and machine–oriented technical resources spent can be in terms of both time and computational cost. Hence, the study will focus especially on deepening the gravitational implications, based on the following facts: *firstly*, the main and longest leg of an interplanetary trajectory corresponds to the cruise phase, in which the spacecraft is essentially affected by a multi-body attraction; *secondly*, when numerically integrating the NBP equation of motion – regardless of whether a computational brute force or a more sophisticated technique is used – the algorithm needs an update of the dynamical states (positions and velocities) of a large number of bodies for each new integration step. Therefore, this tool can help to select which planets must be included, for a given interplanetary trajectory to be planned, neglecting those that do not exert a significant difference in the motion of the spacecraft. It would save time in accessing and downloading from the platform many of the planetary ephemeris, while avoiding long loading times of large data arrays in the workspace of the programming environment. In a society where time and cost prevail, it is hoped that the procedure presented can make a small contribution.

1.3. Objectives of the study

The main goal pursued by this project has already been formalised in the previous section. The following is a summary of some of the specific ones that involve the study in general:

- Understand how the equations of motion of the N-body problem work, especially oriented to its numerical integration

- Introduce the concept of perturbation acceleration threshold as the upper limit magnitude of the vector sum of all perturbing accelerations that a propelled object can admit without significantly affecting its motion. Estimate it for a modern spacecraft

• Obtain and analyse maps representing the control volume under study of the solar system, showing the areas in which a spacecraft, accelerated with respect to the Sun, would be subjected to planetary perturbations above its established acceleration threshold. To develop a criterion that allows identifying the relevant disturbances in a given trajectory

• Validation of the results published on the maps by simulating a real interplanetary ephemeris trajectory, based on the numerical integration of the NBP equations of motion through the application of two of the most widely used special perturbation models: Cowell and Encke

• Comparison of the results obtained with both perturbation techniques

• To apply the knowledge provided by the procedure developed to a real case, consisting of evaluating the feasibility of an asteroid rendezvous with a spacecraft departing from Earth

CHAPTER 2. THE N-BODY ORBITAL MECHANICS

This chapter intends to lay the foundations of the complex dynamic nature of the NBP equations of motion, as well as to delve into the physical meanings of the different terms. A brief subsection will be devoted to the choice of a suitable reference system. Next, an idea will be given of what a perturbation represents and what types exist, to finally develop the general differential equations that govern the motion of a mass particle in a gravitational field created by the rest of the N-1 bodies. In addition, a brief rationale is presented to refer to the way in which these perturbations can be handled and be introduced when applicable.

Once the necessary theoretical part is complete and before describing the structure of the JPL ephemeris model, the concept of *Solar System Barycentre* (hereinafter referred as SSB) will be introduced. The reason lies on that it is recommended to express with respect to it the Cartesian state vectors gathered from the HORIZONS platform, as it is the most reliable coordinate system within the Solar System.

2.1. The N-Body Problem

2.1.1. Choosing a suitable reference system

The first step in this development must be the choice of a "*suitable*" coordinate system with respect to which to refer to the motion or lack thereof. This is not a simple task at all, since any coordinate system has an unavoidable degree of uncertainty regarding its inertial properties[12].

Such an "ideal" frame does not exist in the universe, since a perfect (or absolute) inertial reference frame must be perfectly isolated from any other object within the sidereal space, i.e, infinitely far from every mass particle attracting gravitationally (it is important to remember that reference frames are commonly placed at massive bodies, such as planets or stars).

A simple but rigorous imaginative exercise will mathematically demonstrate a fact that is already widely known: *the more massive a celestial body is, the less accelerated it is* (thus being a better choice as a reference frame). Although it implies to advance a little in the formal development of the N – Body Problem, the common base that they have will allow later to start from a similar concept.

Let us imagine a bounded region of space where *n+1 masses are arbitrarily distributed* (presumably in motion according to the gravitational law). Suppose one of these masses is larger than the others (e.g, the Sun). Consequently, let

[12] An inertial reference frame is characterised by being non-accelerated and non-rotating. Newton defined it as fixed in absolute space, alleging that "*in its own nature, without relation to anything external, remains always similar and immovable*" [11]. However, he failed to indicate how to find such a frame in absolute repose.

us call that large mass M and m_j the rest of n masses, where j = 1, 2, 3 …n. Among the n masses, let us say that n is the largest (e.g, Jupiter). In addition, let us assume that the origin of the frame is located "somewhere" considered as ideally fixed. Thus, the positions of the n+1 masses are perfectly defined and are absolute. The system is illustrated, at an *arbitrary time*, in **Figure 2-1**.

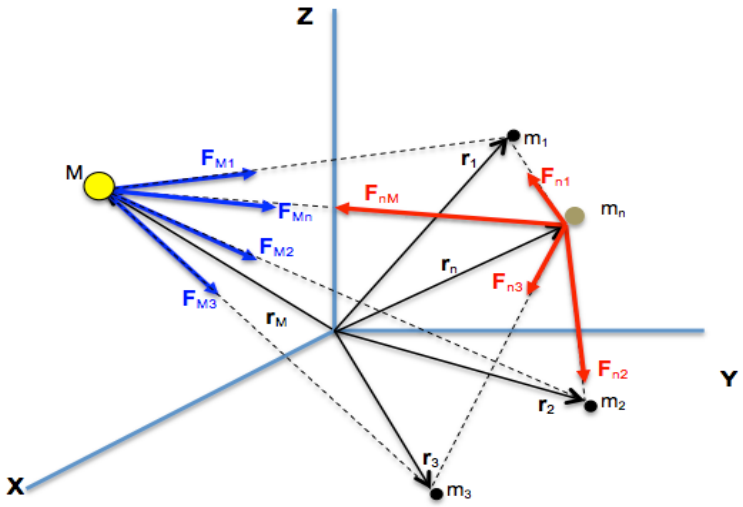

Fig. 2.1 System of n+1 masses, where M is the largest, followed by m_n. The rest are smaller but not necessarily the same. The black arrows correspond to the position vectors; the blue arrows are the forces that M feels from each of n masses while the red ones are the homologous for the n mass.

Following the logical reasoning, let us write mathematical expressions that describe the vector sum force that M and m_n respectively feel at the instant of time reflected in the picture. According to *Newton's Law of Universal Gravitation*[13]:

$$\boldsymbol{F}_M = -G\frac{Mm_1}{r_{1M}^3}\boldsymbol{r}_{1M} - G\frac{Mm_2}{r_{2M}^3}\boldsymbol{r}_{2M} - \cdots - G\frac{Mm_n}{r_{nM}^3}\boldsymbol{r}_{nM} = -GM\sum_{j=1}^{n}\frac{m_j}{r_{jM}^3}\boldsymbol{r}_{jM} \qquad \textbf{(2-1)}$$

$$\boldsymbol{F}_{m_n} = -G\frac{m_nM}{r_{Mn}^3}\boldsymbol{r}_{Mn} - G\frac{m_nm_1}{r_{1n}^3}\boldsymbol{r}_{1n} - G\frac{m_nm_2}{r_{2n}^3}\boldsymbol{r}_{2n} - \cdots - G\frac{m_nm_{n-1}}{r_{(n-1)n}^3}\boldsymbol{r}_{(n-1)n} = -G\frac{m_nM}{r_{Mn}^3}\boldsymbol{r}_{Mn} - Gm_n\sum_{j=1}^{n-1}\frac{m_j}{r_{jn}^3}\boldsymbol{r}_{jn} \qquad \textbf{(2-2)}$$

Where *G* corresponds to the universal gravitational constant; $\boldsymbol{r}_{jM}$ and $\boldsymbol{r}_{jn}$ are the position of the masses *M* and *n*, respectively, relative to the *j* considered mass. According to *Newton's second law of motion*[14], the rate of change of the

[13] Such universal law is widely known and constitutes one of the basic contents of the physics books of any secondary school. Therefore, there is no specific reference worth quoting (it can even be consulted in the Newton's Principia itself, if desired [11]).

[14] Here it applies exactly what has been said for the gravitational law.

momentum of each mass will be proportional to the force impressed and its motion takes place in the same direction as that force. Thus, one can write:

$$\boldsymbol{F}_M = \frac{d}{dt}(M\boldsymbol{v}_M) = M\boldsymbol{a}_M = GM\frac{m_n}{r_{Mn}^3}\boldsymbol{r}_{Mn} - GM\sum_{j=1}^{n-1}\frac{m_j}{r_{jM}^3}\boldsymbol{r}_{jM} \tag{2-3}$$

$$\boldsymbol{F}_{m_n} = \frac{d}{dt}(m_n\boldsymbol{v}_{m_n}) = m_n\boldsymbol{a}_{m_n} = -GM\frac{m_n}{r_{Mn}^3}\boldsymbol{r}_{Mn} - Gm_n\sum_{j=1}^{n-1}\frac{m_j}{r_{jn}^3}\boldsymbol{r}_{jn} \tag{2-4}$$

Where on the right hand side of Equation (2-3) the vector sum has conveniently broken down into two terms and the identity $\mathbf{r}_{Mn} = -\mathbf{r}_{nM}$ has been applied to have the first term in common with Equation (2-4). It is important to note that the equal sign marked in blue is not always true, since certain relativistic effects would result in changes in mass as a function of time. Nevertheless, this effect is not really important now to continue with the proof and can be neglected (furthermore, such relativistic effects are present with more significance as bodies become more massive, as happens with black holes for instance, where the classical mechanical description no longer applies).

If now Equation (2.3) is divided by Equation (2.4) we get:

$$\frac{\boldsymbol{a}_M}{\boldsymbol{a}_{m_n}} = \frac{\frac{m_n}{r_{Mn}^3}\boldsymbol{r}_{Mn} - \sum_{j=1}^{n-1}\frac{m_j}{r_{jM}^3}\boldsymbol{r}_{jM}}{-\frac{M}{r_{Mn}^3}\boldsymbol{r}_{Mn} - \sum_{j=1}^{n-1}\frac{m_j}{r_{jn}^3}\boldsymbol{r}_{jn}} \tag{2-5}$$

The second numerator and denominator terms only differ in the distance between each j mass and the masses M and m_n, respectively. Remembering that the defined space region contains an arbitrary distribution of n+1 masses, nothing opposes to consider that the distances of each mass j (being j = 1, 2, 3…n-1) to M and m_n, respectively, are the same[15] as the distance between M and m_n. So, we can write:

$$\frac{\boldsymbol{a}_M}{\boldsymbol{a}_{m_n}} = \frac{m_n\boldsymbol{r}_{Mn} - \sum_{j=1}^{n-1} m_j\,\boldsymbol{r}_{jM}}{-M\boldsymbol{r}_{Mn} - \sum_{j=1}^{n-1} m_j\,\boldsymbol{r}_{jn}} \tag{2-6}$$

The $\mathbf{r}_{jM}$ and $\mathbf{r}_{jn}$ retain their nomenclature because they exhibit the same magnitude but, since the problem was stated in space, their directions not necessarily have to be the same. Directly from Equation (2-6) it can finally be noted that both summation terms generate two vectors that could differ in their directions but exhibit the same magnitude. In consequence, *the ratio of absolute acceleration vectors will differ to the extent that their masses do*. In this way, the relevant learning from this development finally adopts the form:

[15] It is a valid assumption since the development is based on an imaginative mathematical exercise, which is obviously not likely but not impossible either. The j masses could be aligned while M and m_n be at the same distance from that straight line. Another intuitive way to imagine it would be to consider that the n+1 masses are distributed forming equilateral triangles.

$$\frac{\boldsymbol{a}_M}{\boldsymbol{a}_{m_n}} \propto \frac{m_n}{M} \qquad \textbf{(2-7)}$$

What Equation (2-7) says it is that **the absolute acceleration of a body is inversely proportional to its mass**. Therefore, a reference frame coincident with the centre of mass of the large mass M will exhibit better inertial qualities than its counterpart centred on mass m_n. The outcome of this proof is clear: leaving aside the consideration of relativistic effects, placing a reference system on Earth is better than doing it on the Moon, just as centring it on the Sun is better than doing it on Earth, and so on.

Since the largest body in the Solar System is the Sun, it would not be an irrational decision to place a frame there. However, this work will deal with extremely accurate data achieved from the ephemeris model (see **Section** 2.3 for details) that will require the use of the most inertial coordinate system possible, which is placed at *Solar System Barycentre*. This special frame will be introduced and further discussed in the next section[16] and will be the one with respect to which the position of celestial bodies be expressed when later developing the general equations of motion.

Note: - Even if the assumption $r_{jM} = r_{jn} = r_{Mn}$ had not been made, Equation (2-5) shows that, the only magnitude that can exert a differential factor, is M. Considering the worst case scenario, i.e, M the Sun and m_n the largest planet (Jupiter) the mass ratio is of an order of magnitude of 10^3. So, the only planet that could hypothetically "compromise" the fact that $a_M << a_{mn}$ is a planet that is *massive, is as closer as possible to the Sun and as far as possible to Jupiter at the same time* (so that the summation contributes significantly in the numerator but not in the denominator). A planet that satisfies these conditions is Mercury, which is not massive at all ($M_S/M_{Mercury} \approx 6x10^6$).

2.1.2. External and internal forces. Main sources of perturbation

Before delving into the development of the general equations that govern the motion of objects in space, it is interesting, first of all, to make an exhaustive examination of the perturbations coming from the external and internal forces to which an artificial object may be subjected, leaving aside by the time being those coming from gravitational[17] forces.

It is worth pointing out – albeit a little beforehand – that the concept of *perturbation* arises to counteract its antagonist, the *ideal* relative motion between two bodies. Indeed, within the Solar System, the Sun is the massive body that creates the strongest gravitational field, thus any object is subject to

[16] There it will be seen how actually this reference frame is not absolutely at rest due to its own definition. Nevertheless, its motion takes place at approximately constant velocity, so it is the best way to achieve an "inertial" reference within the Solar System.

[17] Without expressly calling them "*perturbations*", they have already been implicitly introduced in the mathematical development performed in the previous subsection. In fact, the "portrait" shown in **Figure 2-1** and the concepts introduced will be used in the later development of the equations of motion.

its influence. In the absence of other bodies and external or internal forces affecting its motion, an artificial object would simply follow the simpler version of Newton's gravitational and motion laws, respectively[18].

Returning to the picture of an arbitrary moment in which several gravitational masses act on the body within its path, it may also be experiencing other forces; some *internal* (produced by the artificial object itself) and others *external* (coming from sources other than those of a gravitational nature). A typical classification is made without taking into account their origin, because they will be considered as perturbations to all intents at the time of developing the equations of motion. Among the most relevant and commonly managed are:

• **Propulsion/Thrust**: - The body can be a rocket, a spacecraft or even a propelled satellite, thus expelling mass to produce thrust (i.e, propellants). In consequence, based on Newton's third law, an opposing reaction force is expected to balance the total momentum of the system.

• **Atmospheric Drag**: - The body may be moving in an atmosphere where the density of the fluid crossed exerts an aerodynamic resistance (typically known as *drag*) thus applying a force that tends to reduce its acceleration in the direction of its velocity. This effect is relevant when the body moves within an atmosphere (launch and re-entry) or it simply orbits a celestial body at an altitude where the density of the fluid is significant (i.e, usually planets with thermo-fluid dynamic activity). A common threshold for considering when it is relevant in orbits around the Earth is 800 km of altitude[19].

• **Solar Radiation Pressure (SRP)**: - The body may be subjected to a force (for or against) due to the impact of solar radiation on its different surfaces. It is generally small, but can still have a considerable effect, especially when the object exhibits a large area/mass ratio. For orbits around the Earth, the SRP is assumed to be stronger than drag above 800 km of altitude.

• **Non – sphericity of celestial bodies**: - It corresponds to a perturbation to correct the general equations of motion once developed. The reason lies in the fact that, applying Newton's gravitational law requires assuming perfect geometric spheres where the mass is evenly distributed in concentric spherical layers. However, this does not happen in the real nature and the effects of gravity are no longer uniform, beginning to vary within the point-to-point volume. The magnitude of this force becomes strong when the object approaches or orbits the celestial body. In the case of the Earth, it is flattened at the poles and bulged at the equator, and a satellite orbiting at about 370 km of altitude will experience a relative acceleration of the order of 10^{-3}g [12].

[18] It corresponds, by definition, to a 2BP. Indeed, although this document will focus on the development of the general equations of motion and later simplify when needed, it is possible to derivate it directly.

[19] It is a basic concept, but just remember that the term "*altitude*" carries implicitly the assumption that the distance is measured between the object and the selected reference point of the celestial body (for Earth such a reference point is sea level).

• **Magnetic fields**: - Probably the least common effect, it is due to a hypothetical electromagnetic coupling between an electrically charged body (e.g, in satellites the diverse equipment on board or the payload) and the magnetic field of the celestial body, thus causing a deviation force.

As will be indicated in the following subsection, the general equations of motion are commonly obtained without accounting any perturbation other than the arisen from the forces of gravity exerted by the n masses on the i^{th} body whose motion is object of study. Thanks to the development of a vector expression in terms of accelerations, any type of perturbation relevant in a given system can be added later in the equations. Since this thesis deals with machine–oriented numerical integration of the complete equations of motion using *special perturbation methods* (to be discussed in the next chapter) at this point it has been considered convenient to introduce some *analytical formulations* to acquire a background about how the main perturbations work, especially for inclusion in those methods, if apply.

Propulsion/Thrust

The thrust can be easily handled quite directly by resolving the thrust vector into their Cartesian coordinate directions. Thus, the analytical expressions for the perturbing acceleration vector that can be added to the general equations of motion if it becomes relevant are the followings:

$$\boldsymbol{a}_{thrust} = \ddot{x}\boldsymbol{I} + \ddot{y}\boldsymbol{J} + \ddot{z}\boldsymbol{K} \quad where: \quad \ddot{x} = \frac{T_x}{m} \; ; \; \ddot{y} = \frac{T_y}{m} \; ; \; \ddot{z} = \frac{T_z}{m} \tag{2-8}$$

Where T_x, T_y and T_z correspond to the components of the applied thrust and m corresponds to the mass of the artificial body or vehicle. It is important to bear in mind that the thrust can come from an external force – which can already be known or calculated from the analysis of the mechanical system through the suitable parameters – or from an internal force due to the propulsion of the vehicle itself. In the latter case, a further specific analysis should be made since physics changes depending on the type of propelled vehicle (e.g, a *rocket with chemical*, *solid* or *hybrid* propulsion, an *electrically* or *magnetically* propelled spacecraft, using a nuclear power plant…etc.).

In this work, the procedure to be developed to achieve the proposed objectives will require considering electric propulsion. Further details are given in **Chapter 5**, *Estimation of the perturbing acceleration threshold*.

Atmospheric Drag

The equations are riddled with uncertainties about atmospheric fluctuations, the frontal areas of orbiting object (if not constant), the drag coefficient and other parameters. A fairly simple formulation can be consulted if desired in NASA SP-33 [13] but will not be described here since, as mentioned above, this work will focus especially on the study of perturbations within interplanetary trajectories and, in particular, in their cruise phase, where the atmospheric drag *no longer apply*.

Solar Radiation Pressure

The forces due to the SRP rise when photons from the Sun strike the surfaces of the body (i.e, a satellite or spacecraft) and are absorbed or reflected, thus transferring the photon impulse. Contrary to drag, the SRP force does not vary with altitude, its main effect being a slight change in both eccentricity and longitude of the periapsis of the orbit. Its effect is mainly noticeable on satellites that expose large solar panels, such as communication satellites and GPS. Vallado [14] provides the next analytical expression to get the perturbative acceleration vector:

$$\boldsymbol{a}_{SRP} = -\rho_{SR}\frac{c_R A_{Sun}}{m}\frac{\boldsymbol{r}_{sat-Sun}}{r_{sat-Sun}} \quad \textbf{(2-9)}$$

Where:

ρ_{SR} Is the incoming solar pressure, which depends on the time of the year and the intensity of the solar output. It is derived from the incoming solar flux and values about 1358 – 1373 W/m^2 are common.

C_{SR} Is the coefficient of reflectivity that indicates the absorptive or reflective properties of the material, and thus susceptibility to incoming solar radiation. (So, it depends on the specific material of each surface).

A_{Sun} Is the cross-sectional area exposed to the Sun, which changes constantly (unless there is a precise attitude control system, or it is spherical).

M Is the mass of the spacecraft or satellite, which is generally constant but thrust, ablation and other abrasive physical phenomena can remove superficial material and vary this value.

$\mathbf{R}_{sat\text{-}Sun}$ Is the satellite relative to Sun position vector

Note: - As one would probably guess, complex nature can seldom be contained in such a simple expression. And he would be correct, aspects such as the model assigned to attitude control (if it exists), the differing shadow models, accounting for seasonal variations in solar pressure or the treatment for the light-time travel from the Sun to the S/C are challenges that, however, are completely beyond the scope of this work.

Non – sphericity of celestial bodies

The equations of Newton uses the concept of gravitational potential but simplified at maximum. However, if the *potential function*, ϕ, is known, the perturbative acceleration vector can be found according to this expression:

$$\boldsymbol{a} = \nabla\phi = \frac{\partial\phi}{\partial x}\boldsymbol{I} + \frac{\partial\phi}{\partial y}\boldsymbol{J} + \frac{\partial\phi}{\partial z}\boldsymbol{K} \quad \textbf{(2-10)}$$

If considering the case probably more relevant, the Earth, such potential function can be obtained through:

$$\phi = \frac{\mu_e}{r}\left[1 - \sum_{n=2}^{\infty} J_n \left(\frac{r_e}{r}\right)^n P_n \sin L\right] \quad \textbf{(2-11)}$$

Where:

μ_e is the gravitational parameter of the Earth

J_n are coefficients to be determined by experimental observation

r_e is the equatorial radius of the Earth

P_n are the Legendre polynomials

r is the distance to Earth's centre (origin of the potential field)

L corresponds to the geocentric latitude

sin L = z/r (with z a Cartesian component)

There have been various determinations of the J coefficients, which are slightly in variance. Usually the summation is considered until the 7th term, but $J_2 = (1082.64 \pm 0.03)\cdot 10^{-6}$ is clearly the predominant and Equation (2-11) is truncated in the second term (values given by Baker [15]).

It is important to say that, in the potential function written, only the *zonal* harmonics have been considered in first instance, which are those harmonics that are dependent only on mass distribution about the north-south axis of the Earth (dependent on *latitude only*). There exists also *sectorial* harmonics (dependent on *longitude only*) and *tesseral* harmonics (dependent on *both* latitude and longitude). A general expression to account for all three classes of harmonics in the potential function is given by Baker. Thus, retaining until second term in Equation (2-11) and computing its gradient through Equation (2-10) the final expression for the perturbative acceleration vector is[20]:

$$\boldsymbol{a}_{\boldsymbol{J_2}} = -\frac{3}{2}\mu_e J_2 \frac{r_e^2}{r^5}\left[\left(1 - 5\frac{z^2}{r^2}\right)(x\boldsymbol{I} + y\boldsymbol{J}) + \left(3 - 5\frac{z^2}{r^2}\right)z\boldsymbol{K}\right] \quad \textbf{(2-12)}$$

One important thing is that $\mathbf{a}_{J2}$ has been formulated in the *geocentric equatorial coordinate system.* Thus, care must be taken with Earth's rotation if the J_2 effect is applied for Earth–centred orbits. In addition, if $\mathbf{a}_{J2}$ is introduced as perturbation in the general equations of motion when numerically integrate for an interplanetary trajectory (as would be the case) a transformation to the Heliocentric – Ecliptic coordinate system must first be performed.[21]

[20] The maths involved in the development are not excessively complex, but require first calculating the Legendre polynomial, P_2, which is given simply by the expression: $P_k = \frac{1}{2^k k!}\frac{d^k}{dz^k}(z^2-1)^k$. For the sake of simplicity, the intermediate steps have directly been skipped. All calculations can easily be followed in the Baker reference [15].

[21] Actually, for later analyses of discrepancies of the simulated trajectories with respect those obtained from the ephemeris, positions and velocities must be expressed in the SSB, and this will also be carried out through the corresponding transformations (more on this later).

2.1.3. General differential equations of motion

For this development we shall assume a system of n-bodies (m_1, m_2 … m_n) one of which is the body whose motion is described, let us call it the i^{th} body, whose mass will be referred to as m_i. An advantage of the above discussion on the choice of a suitable reference coordinate system is that it is now possible to start from the **Figure 2-1** and one of the equations formulated above. To do this, however, a "visual exercise" must be done to change that "portrait" a little:

1) The large mass *M is not considered now*. Therefore, all the blue arrows and the red one called F_{nM} no longer apply.

2) For the rest of the n masses, let us consider that m_n *is now* m_i (the i^{th} body) and m_3 *is now* m_n (the n^{th} mass).

3) The origin of the reference frame changes its denomination of "somewhere" to "known place". That chosen location is none other than the SSB, according to what was stated at the end of the discussion of a suitable frame.

As mentioned in the previous Subsection, for the development of the general equations of motion the presence of perturbations is discarded beforehand to avoid densifying the notation. Once obtained, it will be clearly indicated how to introduce them in case of need. Departing from Equation (2-4) applying the changes mentioned in the notation and not accounting relativistic effects (the masses do not change) we have:

$$\boldsymbol{F}_{m_i} = \frac{d}{dt}\left(m_i \boldsymbol{v}_{m_i}\right) = m_i \boldsymbol{a}_i = -G m_i \sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{m_j}{r_{ji}^3} \boldsymbol{r}_{ji} \tag{2-13}$$

Which, mathematically expressed in a more convenient fashion:

$$\frac{d^2 \boldsymbol{r_i}}{dt^2} = \ddot{\boldsymbol{r}}_i = -G \sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{m_j}{r_{ji}^3} \boldsymbol{r}_{ji} \tag{2-14}$$

Where G is the universal gravitational constant and $\boldsymbol{r}_{ji}$ is the position of the i^{th} mass relative to each m_j.

Even having made the assumptions of neglecting relativistic effects, having assumed an unpowered flight ($dm_i/dt = 0$, i.e, the i^{th} body does not expel mass as no thrust is considered) and having neglected any other type of perturbation, Equation (2-14) is a *second order, non-linear, vector, system of differential equations*, which still has defied solution in its present form.

The development is not yet completed. The acceleration vector describing the motion of the i^{th} mass under study is currently expressed with respect to an *inertial fixed reference coordinate system*, which despite being valid for performing numerical propagations of initial conditions (position and velocity) to define its absolute motion within the N-body gravitational field, the concept of

perturbation has not yet arisen. Of course, it is already mathematically included in Equation (2-14) but it has not yet been explicitly extracted to allow carrying out the studies and analyses that the present work is aimed to.

If the terms of Equation (2-14) are now rearranged to describe the relative motion between two bodies, with the remaining n-2 bodies acting as perturbations – what is commonly called as Kepler's perturbed problem – it is possible to write:

$$\ddot{\boldsymbol{r}}_1 = -G\sum_{j=2}^{n}\frac{m_j}{r_{j1}^3}\boldsymbol{r}_{j1} \tag{2-15}$$

$$\ddot{\boldsymbol{r}}_2 = -G\sum_{\substack{j=1\\ j\neq 2}}^{n}\frac{m_j}{r_{j2}^3}\boldsymbol{r}_{j2} \tag{2-16}$$

Where the body represented by 1 typically is a large mass dominating the gravitational field (e.g, the Sun, as is the case in the cruise phases of interplanetary trajectories) and m_2 is the small mass whose motion is totally influenced by the large mass and the remaining m_3, m_4 … m_n, typically being the planets and natural satellites of the planetary system (in the present case, the Solar System).

It is now possible to express the motion of mass 2 relative to mass 1:

$$\ddot{\boldsymbol{r}}_2 - \ddot{\boldsymbol{r}}_1 = \ddot{\boldsymbol{r}}_{12} = -G\sum_{\substack{j=1\\ j\neq 2}}^{n}\frac{m_j}{r_{j2}^3}\boldsymbol{r}_{j2} + G\sum_{j=2}^{n}\frac{m_j}{r_{j1}^3}\boldsymbol{r}_{j1} \tag{2-17}$$

Expanding now the first term of each summation terms:

$$\ddot{\boldsymbol{r}}_{12} = -\left[G\frac{m_1}{r_{12}^3}\boldsymbol{r}_{12} + G\sum_{j=3}^{n}\frac{m_j}{r_{j2}^3}\boldsymbol{r}_{j2}\right] - \left[-G\frac{m_2}{r_{21}^3}\boldsymbol{r}_{21} - G\sum_{j=3}^{n}\frac{m_j}{r_{j1}^3}\boldsymbol{r}_{j1}\right] \tag{2-18}$$

Finally considering again the identity $\mathbf{r}_{12} = -\mathbf{r}_{21}$ the first terms in each bracket can be combined to reach the pursued expression:

$$\ddot{\boldsymbol{r}}_{12} = -G\frac{m_1 + m_2}{r_{12}^3}\boldsymbol{r}_{12} - \sum_{j=3}^{n}Gm_j\left(\frac{\boldsymbol{r}_{j2}}{r_{j2}^3} - \frac{\boldsymbol{r}_{j1}}{r_{j1}^3}\right) \tag{2-19}$$

Equation (2-19) represents the acceleration of m_2 (i.e, the object of interest, usually a satellite or a spacecraft) *relative* to m_1 (the dominant body, usually the Sun for interplanetary computations). The first term corresponds to the direct application of *Newton's Law of Universal Gravitation* between two bodies. The effect of the second term is to account for the perturbing effects exerted on the motion of m_2 by the rest of the masses. It is worth mentioning that this term is, in turn, composed of the two terms within the parenthesis of the summation:

- The first term is usually referred to as the *direct term* and corresponds to the gravitational attraction between each perturbing body and m_2.
- The second term is usually referred to as the *indirect term* and corresponds to the gravitational attraction between each perturbing body and m_1.

Apart from this, as indicated earlier, any type of perturbation relevant in a given system can be added directly into the equations as an additional term, which can be generalised as $\mathbf{a}_p$ as follows:

$$\boldsymbol{a}_p = \boldsymbol{a}_{thrust} + \boldsymbol{a}_{drag} + \boldsymbol{a}_{SRP} + \boldsymbol{a}_{non-sphericity} + \boldsymbol{a}_{other} \quad \textbf{(2-20)}$$

The last term called "*other*" is introduced to account for any additional model considered relevant for the study case (such as *albedo*, *tides* or *magnetic fields*, among others). The reason why Equation (2-20) is introduced separately lies in the fact that – besides the mentioned simplicity of handling – among the physical forces affecting the motion of a satellite or spacecraft, by far the largest is due to *gravitation*, usually followed by *atmospheric drag*, *SRP* and several others such as *tides* (according to some analysis carried out by Vallado [16]).

2.2. The Solar System Barycentre

Having reached these lines, the concept of SSB has already been mentioned at least twice. There exist many different frames and each of them exhibits its strengths (and weaknesses) depending on the particular system under study. For instance, some are useful to describe states on the surface of the Earth, Moon or any other celestial body; others are useful to describe the relative geometry between the Sun, Earth and/or Moon. However, as said before, a coordinate system is inertial only when it is not accelerating. When referencing motion in the Solar System, the only "truly inertial" coordinate system that can be found is, according to what *Parker and Anderson* affirmed in their document [3], one that is not rotating and centred in the Solar System barycentre. Strictly speaking, no Earth-centred coordinate system can be inertial, not even one that is not rotating, since the Earth is accelerating its orbit as it revolves about the Sun. The same applies – to a lesser extent obviously – to the Sun, its orbit accelerates as it revolves about the likely massive black hole that occupies the centre of our galaxy (the *Milky Way*). Therefore, neither the Geocentric–Equatorial nor the Heliocentric–Ecliptic coordinate systems can be strictly referred as inertial. Instead, they are merely *nonrotating*[22].

Without prejudice to what has been said, a question is still pending: *What is the Solar System Barycentre? What does it consist of physically?* Being quite intuitive, the mathematical concept of barycentre already allows guessing to a greater or lesser extent what it consists of for anyone who is not profane in

[22] When later describing the JPL ephemeris model and HORIZONS platform, brief guidelines will be given on the management of data expressed in the SSB frame to be used as initial conditions in the general equations of motion given by Equation (2-19).

science in general. Although it is not an objective of this thesis to delve excessively into it, a brief explanation is introduced given the need of its use.

The Solar System is ultimately a discrete system, as it is composed of one star (the Sun) and planets, natural satellites, comets, asteroids and other celestial identities. Thus, it has a *centre of mass* that, at any given instant, can be calculated simply by applying the following expression:

$$\boldsymbol{r}_{SSB} = \frac{\sum_{i=1}^{n} m_i \boldsymbol{r_i}}{\sum_{i=1}^{n} m_i} \qquad \textbf{(2-21)}$$

Where m_i and $\mathbf{r}_i$ are the masses and instantaneous positions, respectively, of each celestial body (let us say those most important, i.e, the sun and its planets) with respect to a given arbitrary origin of reference[23]. Such a reference can be placed in the centre of the Sun. Then, $\mathbf{r}_{Sun} = 0$ and the calculations are facilitated since the Sun no longer needs to be considered[24]. Indeed, the masses aliens to the Sun tend to exert a gravitational pull in its centre of mass (as the Sun does on them, it is philosophically "*a massive game of tug of war*"). If the positions of the rest of the n-1 masses are approximately balanced in that moment, the actual position of the CM will be very close to Sun's centre. Instead, if the planets are aligned, such a position may even be outside the Sun's surface (but never outside the Sun's corona).

Since the positions of the n-1 masses with respect to the Sun are dynamic[25], it is simple to guess that the position of the CM with respect to the Sun's centre will vary (slowly, since the Sun is by far the largest contributor). Finally, it leads us to the idea that both, the planets and the Sun are actually revolving about that abstract dynamic point. When the CM is placed outside the solar surface, the Sun experiences a *translational* motion. Instead, when the CM lies inside, it experiences a motion known as "*wobble*". **Figure 2-2** shows a picture of the positions exhibited by the CM (SSB) over time. It is currently outside the solar surface.

Appendix A.1 includes some figures obtained from a simulation that allows observing the orbit of the Sun about the SSB, as well as to examine the relevance of each planetary effect. Finally, two things can be pinpointed:

1) The chart can be obtained with a simple code that gathers ephemeris of the positions of the planets and the Sun (e.g, in SSB) and implements Equation (2-21) for a given time interval (i.e, years).

[23] As can be seen in any book of mechanics, the choice of the origin of a reference system does not affect the calculation of a centre of mass in either a discrete particle system or a continuous material. Having said this, it is usually chosen some convenient reference (e.g, a corner in a mechanical piece, the large particle in a particle system…etc) to facilitate the calculations.

[24] Note that this means that one of the n terms in the numerator will be zero. However, the same is not true for the denominator, where the term associated with its mass is taken into account.

[25] This does not mean that the position of the Sun is fixed at all. It has only been mentioned because the origin of the reference remains at its centre. So, the movement is relative to it.

2) The SSB is ultimately a fairly good point for locating the origin of a coordinate system due to its inertial qualities. Of course, the barycentre itself is in motion within the Solar System, which revolves about the galaxy in turn. Nevertheless, its motion can be approximated as constant and rectilinear.

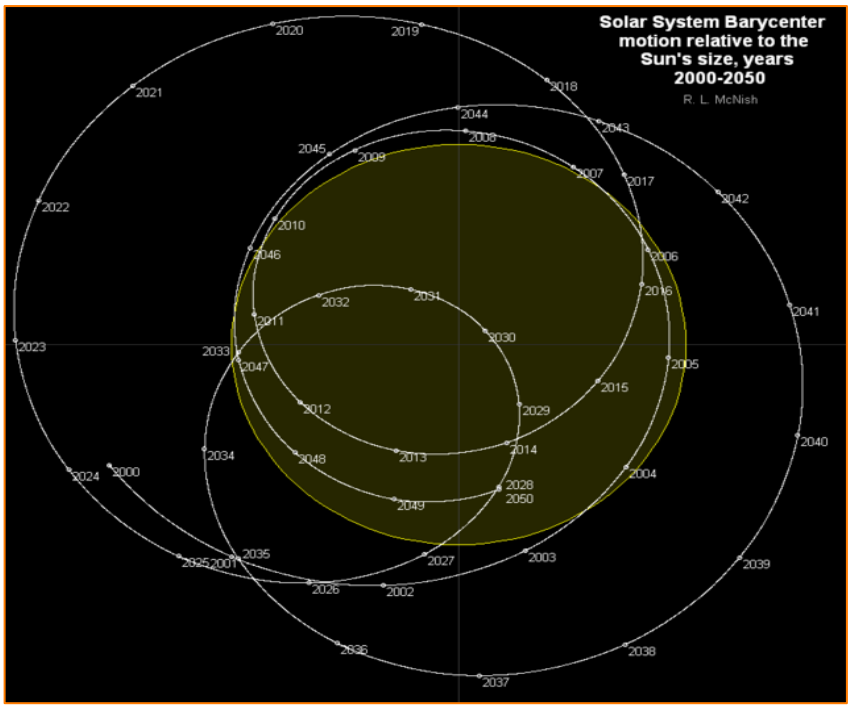

Fig. 2.2 SSB chart. Courtesy of R. L. McNish. Calgary Centre of the Royal Astronomical Society of Canada.

2.3. JPL Ephemeris model

This section presents the JPL ephemeris model, which is the cornerstone of this work. This model provides *high-precision ephemerides* in support of spacecraft navigation and other activities related to Solar System bodies. These ephemerides are obtained by numerical integration of the full[26] dynamical NBP model written in Equation (2-19). It is implemented in *HORIZONS*, a web tool-built that is constantly being updated by the JPL. Available options include the *Sun*, *planets*, *comets*, *asteroids*, *natural satellites*, *spacecraft trajectories* and several *dynamical points* such as the Earth-Sun L_1, L_2, L_4, L_5 and *systems barycentre's*.

As discussed in the motivation part, the aim is to deepen the understanding of the influence that perturbations actually exert in the trajectory calculations, with particular emphasis on those interplanetary carried out by modern spacecraft. Because there does not exist two equal trajectories, a tool/procedure will developed and validated that can help detect, for a given

[26] It means including all relevant perturbations. As a reminder: *thruster firings*, *solar pressure*, *extended spherical harmonic gravity fields*, *atmospheric drag*, and whatever other dynamic model is used for navigation.

one[27], which perturbations are not relevant and can thus be neglected without a significant loss of accuracy. In this framework, the JPL ephemeris model (through its platform) will serve as a benchmark[28] to perform the necessary analyses to fulfil the objectives stated in chapter 1.

Previously, it was mentioned that the ephemeris data that provided by the JPL come from an accurate time–integration that takes into account a large variety of phenomena that could affect the dynamical model (basically everything that could be considered). This is not entirely true; the word "*ephemeris*" has a Greek origin, which means diary or calendar and is a highly accurate record of the states of the celestial bodies with significant mass within the Solar System. In this sense, the information also comes from very precise observations made in various sites on Earth, or from orbiting satellites, among others. However, this assertion was made because the source from which this work is fed corresponds to the computed ones.

There exist a wide variety of ephemerides and they are classified mainly by *type*, *accuracy* and *time span*. The latest model containing planetary and lunar ephemerides is called DE430, which takes in consideration the friction damping between the fluid core and the elastic mantle of some celestial bodies. Nevertheless, due to that damping term, these ephemerides are not suitable for extrapolation beyond several centuries into the past. To cover a longer time span (to the detriment of accuracy) the ephemeris model DE431 was obtained without it. Detailed information can be consulted in Folkner et al. [16].

Some of planetary ephemerides are released saved in files of Chebyshev polynomials fit to the Cartesian positions and velocities of the planets, Sun and Moon commonly exhibiting an interval of 32 days [4]. The positions are integrated in AU[29] but with the polynomials saved already in km, and the integration time is carried in days of Barycentric Dynamical Time (TDB).

In case of accessing the ephemeris by means of an automated tool developed for that purpose (self-made or using for instance one of the references provided by the JPL, Standish E.M [17] that runs in *MatLab*) it will carry out all the necessary steps to finally offer the information in a useful way for the user (i.e, access via a secure ftp server placed on the NASA and JPL ports, convert to binary files the 20-years ASCII blocks on where ephemeris are released and carry out the interpolation of the data). The input is just a header containing the main information of the ephemeris used (in our case DE431) and the ASCII itself. The program then interpolates for an epoch specified in *Julian*

[27] The purpose of this work is not to perform accurate simulations by refining simpler models (as 2BP, CRTBP, patched-CRTBP…etc) with high-fidelity ephemerides nor to compare performances between simulations using different perturbation techniques. As indicated, a wide bunch of works focussed in those directions can be found in the literature. Instead, although these special perturbation techniques will be used for validation, the main objective is to define a procedure that can be applied systematically to any desired trajectory to perform simulations that include only the necessary perturbations needed in order to not lose significant accuracy.

[28] In **Chapter 4**, further detail will be given to formally establish the strategy for developing the tool/procedure that will enable the desired analyses to be carried out.

[29] The exact value used is *1 AU = 149597870.700 km*, as adopted by the IAU in 2012.

days, using the Chebyshev polynomials cited above and a recursive formula to ultimately obtain the state vectors of the desired celestial bodies.[30]

In case of not using an automated tool, it is possible to manually access the HORIZONS platform and setup all the parameters to get the desired ephemeris. This option offers less customisation regarding the time span and some other parameters, but is preferred in this work because there is no need to perform a continuous gathering of ephemeris. Rather, only one set of ephemerides containing planets, the Moon, the Sun and one spacecraft is needed. An advantage is also that dates can be entered directly into standard *Gregorian calendar days* instead of the *Julian Days count*. The files will be generated as *.txt* that the user must handle properly due to the fact that the state vectors data are embedded among a lot of other information. To do so, some *MatLab* functions have been created to perform proper reading and loading when the ephemerides are needed for the main script or other functions. They can be consulted in **Appendix B.1** if desired.

Finally, before describing the platform that serves as an interface for the user, it is worth mentioning that the ephemeris model used in this work, among all those available, was the ***DE431***. It was created in April 2013 and includes librations and 1980 nutation. It is referred to the IRF, version 2.0. It covers the time span from *Julian Ephemeris Day* (JED) -3100015.5 (15th August 13200 BC) to JED 8000016.5 (15th March 17191 AD)[31].

2.3.1. HORIZONS platform

Having introduced the main features of the model created by the JPL to provide high-fidelity ephemerides, now is the time to briefly describe the platform from which the necessary data has been achieved.

The JPL HORIZONS on-line solar system data and ephemerides computation service provides access to key Solar System data and flexible production of highly accurate ephemerides[32] for solar system objects (*829521 asteroids*, *3594 comets*, *210 planetary satellites*, *8 planets*[33], the *Sun*, *L1*, *L2*, *selected spacecraft*, and *system barycentres*). HORIZON is provided by the Solar System Dynamics Group of the Jet Propulsion Laboratory. Essentially, the platform offers **five types** of customizable output:

[30] Internal processes concerning access to server ports and handling of data up to the final version provided to the user are out of the scope of this work. Nonetheless, some of the above references constitute a good starting point for further knowledge if desired (*Newhall* [4], *Folkner* [16] and *Standish* [17]).

[31] Here it can be noted the long time span with which the DE431 was gifted with after the DE430.

[32] The last version corresponds to the 4.50 and it was updated on 16 August 2018.

[33] Although the IAU recently removed Pluto from the list of planets, it is included with the planets.

1. **Observables**[34](RA/DEC, Az/El, physical aspect, angles, uncertainties…etc)
2. **Osculating orbital elements**
3. **Cartesian states vectors**
4. **Close approaches to planets** (and 16 largest asteroids)
5. **SPK**[35] **binaries trajectory files** (asteroids and comets only)

Among the available options, the needs of this work correspond to **1** and **3**. Some observables[36] are used to acquire physical properties of the bodies that will help in later analyses, plots or animations (e.g, recreating the real motion of the Solar System scenario in the time window of the simulation performed).

Before describing the set of parameters that must be established for individually getting the ephemerides, some of the most relevant limitations warned by the platform and that directly affect the development of this work are pointed out[37]. The first four are generalist while the last one particularly affects interplanetary trajectories:

- To produce an ephemeris, observational data containing measurement errors are combined with dynamic models containing modelling inaccuracies. The best fit is developed to statistically minimize those errors. The resulting ephemeris has an associated uncertainty that fluctuates with time. Due to this, uncertainties in major planets range from 10 cm to several hundreds of km in the state-of-the-art JPL/DE431 ephemeris.
- Cartesian state vectors are output in *16 decimal places*. This does not mean that all digits are physically meaningful. The full precision may be of interest to those who study the ephemeris or as a source of initial conditions for later integrations.
- Solar relativistic effects are included in all the dynamics of planets, lunar and small bodies, excluding satellites.
- Deviations due to other gravity fields can have a potential effect on the level of 10^{-4} arc seconds, but are currently not included here. Satellites from other planets, such as Jupiter, could also experience deviations at the 10^{-3} arc second level.

---

[34] RA and DE correspond to the Right Ascension and Declination, respectively (used for the Right Ascension–Declination system when some astronomic studies need to project the position of a celestial object against the celestial sphere). Az/El corresponds, by its part, to the Azimuth and Elevation (used in the Topocentric coordinate system).

[35] The SPK files can be used by existing visualization, animation and mission-design software.

[36] This is the case of *masses*, *radii* and *sidereal orbit periods* basically (if any other magnitude is needed for a punctual auxiliary test, it is taken in isolation)

[37] This and much other information can be obtained from the HORIZONS user manual, which is constituted by tens of pages plenty of detail. It can be found at the URL: https://ssd.jpl.nasa.gov/?horizons_doc#purpose

- For interplanetary spacecraft, users having high-precision applications should contact JPL Solar System Dynamics to verify the status of the specific trajectory in HORIZONS, since the platform will always have the latest comet/asteroid/natural satellite solutions, but keeping current with the externally produced spacecraft trajectories is problematic; there is no mandate or funding or staff for this, and manoeuvres and *mission planning changes can occur without notification*.

There are some reflections to be made at this point, although appropriate rationales will be given later when the tools are developed or the results analysed, if apply. The first important question concerns the second point. *MatLab* – the programming environment where all implementations will take place – works with exactly that number of decimals in its double arithmetic precision. In the next chapter, a section will be introduced describing the main integration schemes and their errors. There it will be discussed that, when integrating numerically, any type of numerical method is not error-free. Therefore, the need arises to use initial conditions with extra significant decimal positions[38] with respect to those are actually needed, depending on the number of steps taken by the integration[39].

On the other hand, the last limitation warns that, if the objective were to perform high-precision simulations by numerical integration of the equations of motion, even using the best numerical schemes and the most complex models to account for all relevant perturbation in an interplanetary flight, a significant discrepancy would be guaranteed, due to the uncertainties inherent in the ephemerides themselves (e.g., dates and firings for manoeuvre corrections, changes in a mission in course[40] or the inclusion of models to emulate tides, albedos or other particular phenomena not considered in general terms).

Leaving this discussion closed for the time being, the last question related to this subsection is to briefly describe the basic mechanism of the HORIZONS interface, especially focussed on our needs. The platform offers six main blocks of settings that can be chosen. The user is requested to define the following required information:

- **Ephemeris Type:** - Vector Table is chosen, since it allows generating a Cartesian state vector table of any object with respect to any major body.

- **Target Body:** - The platform shortlists the most frequent celestial bodies and a finder for specific selections. The needs of this work are focussed on the Sun, the 9 planets of the Solar System, the Moon and the planned trajectory for BepiColombo.

[38] That will not be possible, based on the limitation alerted by HORIZONS.

[39] A formula will be given as a guide to estimate approximately how many extra digits are needed as a function of the number of steps in the integration process.

[40] This is the case of the interplanetary mission selected for the validation, which will be based on the BepiColombo mission. The rationale for this will be discussed on **Chapter 5**, *Estimation of the perturbing acceleration threshold*.

- **Observer location:** - If the ephemeris type is set in the option “Observer Table”, a wide offer of different locations on Earth is released. However, when the “Vector Table” is previously selected, the own platform sets up the SSB reference frame (which is the recommended for handling Cartesian coordinates)

- **Time Span:** - It can be defined following the structure (Start Time, Stop time, Step Size) where the two first items should be specified calendar dates following the structure YYYY{BC|AD}-MM-DD and optionally times {HH:MM} (if the ephemeris allows it) if Gregorian dates are used or following the structure {JD} DDDDDDD.DDDD if Julian dates[41]. The complement indicated as “BC” or “AD” must be indicated only when apply. Apart from this, each selected target body has its own limitations regarding the time span selectable.

- **Table Settings:** - Request to define a set of output parameters. It offers two reference systems, where the default assigned is the ICRF/J2000.0 required in this case. Regarding the selection of the reference plane (coordinate system) again the default selected is the correct one: Ecliptic and Mean Equinox of Reference Epoch. To avoid mistakes, the output units should be the international (km and km/s). No aberration correction must be selected to get dynamic state vectors (also set by default; This is offered when observations are of interest). Finally, the vector table type for the output selected is type 2 (x,y,z,v_x,v_y,v_z).

- **Display/Output:** - Among the three options available (HTML by default, plain text and download/save) is the latter the most interesting to apply the MatLab functions developed to read and load the ephemerides.

To conclude, the last relevant thing that remains is the establishment of the only one set of ephemerides that is required from HORIZONS in this work. It includes the ***Sun***, the ***nine planets***, the ***Moon*** and ***BepiColombo S/C***. The chosen dates come imposed[42] by the BepiColombo mission (***2018-Oct-21 to 2025-Nov-2***)[43] and the selected time span is 6 hours. This data configuration is valid both for the development of the tool/procedure and for its validation.

2.3.2. Ephemeris data management

This subsection aims to clarify how the ephemerides gathered from HORIZONS will be handled. The dynamic state vectors containing positions and velocities in Cartesian coordinates for each celestial body are obtained by applying *MatLab* functions (presented in **Appendix B.1**) that read and load each of the corresponding ephemeris. With a *time span of 6 hours*, data arrays with dimensions of *10277x3* are obtained for both position and velocities for each celestial body.

[41] Items in { } are optional, but it is recommended to be organised and avoid mistakes.

[42] Actually the first available date is at 02:13 CT and the last at 08:42 CT, but to avoid problems the first data gathered was delayed almost a day while the last advanced almost 9 h.

[43] The total time duration of the mission is 2570 days (61680 hours)

The reason for choosing that time span between ephemeris data responds to the high precision needed to validate the tool/procedure. Two special perturbation techniques (Cowell and Encke) will be applied to Equation (2-19) to compare solutions for the complete and partial cases of a particular interplanetary mission (i.e, considering all bodies in the perturbing term or only those that the tool indicates as relevant). The step size to carry out the numerical integration needs to be synchronised with the dynamic positions of the celestial bodies, as they participate in the perturbing term at each new integration step. Since the simulated trajectory is that of BepiColombo and contains multiple flybys (gravity assists) the state vectors of the planets involved must be little spaced in time to allow accurate detection of when the S/C passes through its Spheres of Influence[44] (SOI). It is crucial because, in that spherical zone, a transition of the dominant body occurs and thus the algorithm needs to change the configuration to continue with the integration for the next step.

On the other hand, the ephemerides loaded in the workspace of the programming environment (*MatLab*) refer to the SSB and the first available state vector corresponds to October 21st, 2018 being the distance to Earth (from where the S/C was obviously launched) about $3.4 \cdot 10^5$ km. This indicates that the S/C is still within the Earth's SOI[45]. Consequently, when initialising trajectory propagation in simulation (which is made only once) the state vectors should be referred to the non-rotating, non-inertial reference frame placed at centre of the major (dominant) body. **Figure 2-3** shows this transformation between the frames. According to the first term of Equation (2-19):

$$\boldsymbol{r}_{12}(t_0) = \boldsymbol{r}_2(t_0) - \boldsymbol{r}_1(t_0) = \boldsymbol{r}_{BepiColombo_{eph_{SSB}}}(1,:) - \boldsymbol{r}_{major_{eph_{SSB}}}(1,:) \quad \textbf{(2-22)}$$

$$\boldsymbol{v}_{12}(t_0) = \boldsymbol{v}_2(t_0) - \boldsymbol{v}_1(t_0) = \boldsymbol{v}_{BepiColombo_{eph_{SSB}}}(1,:) - \boldsymbol{v}_{major_{eph_{SSB}}}(1,:) \quad \textbf{(2-23)}$$

Where, on the right hand side, the subtraction uses the first row of the corresponding ephemeris. The origin of times is $t_0 = 0$.

For its part, the implementation of the perturbation term of Equation (2-19) requires the inclusion of the dynamic positions of the rest of the celestial bodies. This can be easily handled for each new integration step, since according to what has been said before, the step size is synchronised with the time span between ephemeris data, which means that each new integrated state of the S/C will refer exactly to the same epoch. Thus:

$$\boldsymbol{r}_{j2}(t_{i-1}) = \boldsymbol{r}_2(t_{i-1}) - \boldsymbol{r}_j(t_{i-1}) = \boldsymbol{r}_{BepiColombo_{SSB_{|\boldsymbol{eph\ if\ i=1}}}}(i,:) - \boldsymbol{r}_{j_{eph_{SSB}}}(i,:) \quad \textbf{(2-24)}$$

$$\boldsymbol{r}_{j1}(t_{i-1}) = \boldsymbol{r}_1(t_{i-1}) - \boldsymbol{r}_j(t_{i-1}) = \boldsymbol{r}_{major_{eph_{SSB}}}(i,:) - \boldsymbol{r}_{j_{eph_{SSB}}}(i,:) \quad \textbf{(2-25)}$$

---

[44] If the space of time between state vectors is large, there is a risk of not detecting such a cross with sufficient precision, which inevitably leads to an increase in error as a flyby lasts few hours.

[45] SOI radii are time-dependent, according to the orbital motion of each celestial body with respect to the others (especially wrt Sun). The algorithm will take this into account when implementing the special perturbation techniques. However, the Earth's SOI does not change significantly due to the small eccentricity of its orbit. The mean value of its radius is $9.24 \cdot 10^5$ km.

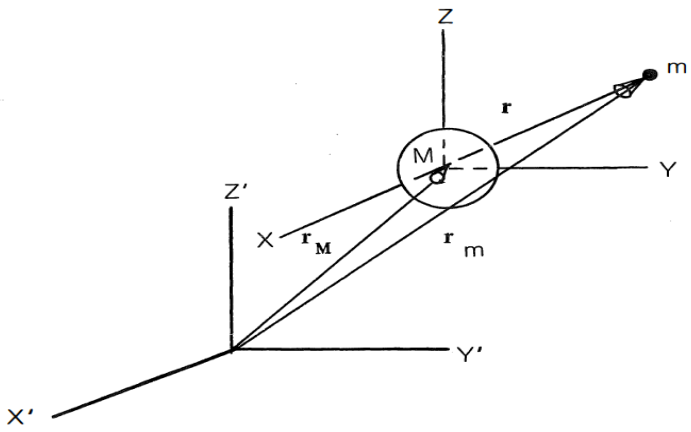

Fig. 2.3 Transformation between SSB "inertial" reference frame [X' Y' Z'] and a non-rotating, non-inertial reference frame with its origin in the central body [X Y Z]. Propagation of trajectory of m (S/C) is made with respect M (the major). ***[Bate, 1971]***

It must be pinpointed several things regarding the previous equations:

1) The index called "i" ranges from i = 1 to the length of ephemeris arrays.

2) The time sub-index is set to (i-1) to match $t_0 = 0$ with i = 1 (there is no "zero" row).

3) For i > 1, the state of BepiColombo does not come from the ephemeris but from the integration itself (Equation 2-24).

4) Perturbations are ordered by an index j (each celestial body always has the same number). For most of the interplanetary trajectory, the S/C is in relative motion about the Sun and the rest of planets and the Moon act as disturbances. However, the major body at the beginning is another (as also happens when the S/C penetrates the SOIs of other planets). Consequently, in those periods of time, the Sun will actually act as another disturbance.

As seen, all the above equations are written with the states (positions and velocities) expressed in SSB, since this is the way in which – according to **Figure 2-3** – relative states can be obtained (i.e., the S/C with respect to the dominant body and both the S/C and the dominant body with respect to each perturbing body respectively). Equations (2-24) and (2-25) always apply before each integration step provides the updated r_{12} (t_i). It is important to note that the application of Equation (2-25) does not require any transformation in the ephemeris participating in it (since the data matrices are already loaded and are not an output, thus they are only an intermediate calculation according to index i). However, in Equation (2-24) the updated position of the S/C must be entered with respect to the SSB, when the integration actually returns its position

relative to the dominant body[46]. Due to this, the propagated state vectors of the S/C must be expressed with respect to SSB again after each integration step.

Figure 2-4 may provide a better understanding. Let the blue circle be the dominant body (i.e., the Earth at the beginning) and the red the object of interest (i.e., the spacecraft). The reference frame has its origin in SSB. At t_0, the position of the S/C relative to the Earth can be calculated by Equation (2-22) where both position vectors come from ephemeris expressed in SSB. Applying the general equations of motion stated in Equation (2-19) for the defined step size, the new relative state of the S/C (brown dashed arrow) is obtained. Thus, in order to obtain the updated position of the S/C at t_1 in SSB – needed to apply Equation (2-24) in the new integration step – Earth's position vector in SSB at t_0 must be added. In the new step, before the integration takes place again, a new S/C position vector relative to Earth is calculated, since the planet has moved (the short arrow in green) and so on for the rest of steps while the S/C is inside its SOI. When shifting of dominant body, the algorithm changes its configuration and applies the appropriate transformations between reference systems following the same routine.

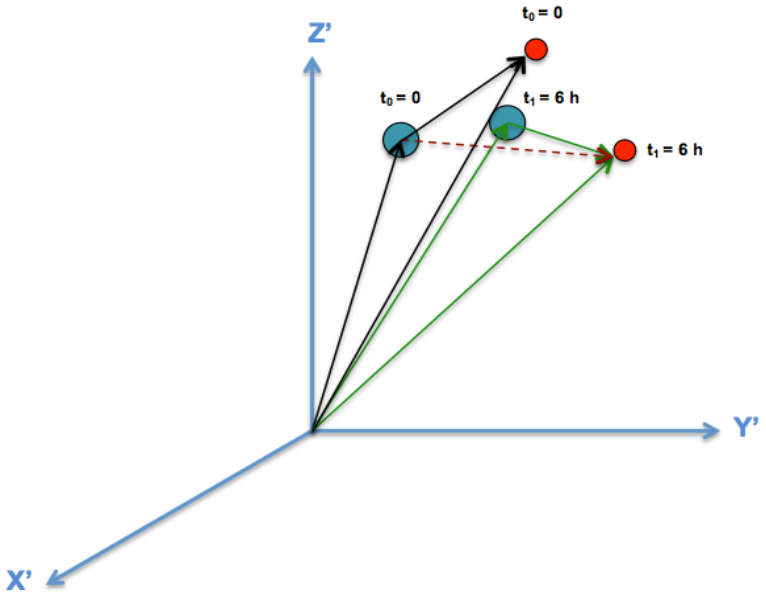

Fig. 2.4 Motion of the S/C relative to the dominant body.

[46] The initial conditions are established – only once – relative to the dominant body. Since the propagated relative position vector is an output of each integration step, the recursion of the process will implicitly tend to return the state expressed in the same conditions.

CHAPTER 3. SPECIAL PERTURBATION TECHNIQUES

The main goal of this work lies on the study of the astrodynamics from the perspective of the effect that perturbations exert in the motion of the celestial bodies, with particular emphasis on an artificial body (an spacecraft) in motion within a complex gravitational field that arises created by them. A procedure to detect when such body can be submitted to perturbations that significantly can modify its trajectory will be presented on next Chapter. Given that such tool needs to be validated and the most accurate way is by integrating the N-Body Problem general equations of motion, this Chapter intends to establish the foundations of two of the most useful machine-oriented methods available, which will be implemented for that purpose: Cowell's and Encke's methods.

3.1. Introduction

The concept of perturbation is quite familiar and, essentially, anyone can guess in a greater or lesser extent what is: a deviation from some normal or expected event. In terms of motion, macroscopically one tends to view the universe as a highly regular and predictable scheme. Yet when it is analysed from accurate observational data, it is found that there seem to be distinct, and times unexplainable, irregularities of motion superimposed upon the average movement of the celestial bodies. Many of the most common sources of perturbations have already been defined and deepened in previous chapter[47].

In the past the existence of perturbations provided great findings (e.g., Neptune was deduced analytically from analysis of the perturbed motion of Uranus and the first accurate prediction of the return of Halley's Comet in 1759 was made from calculation of the perturbations due to Jupiter and Saturn, among many other reported in the literature). Nevertheless, especially in the interplanetary missions, the S/C would miss its target entirely if relevant perturbing effects were ignored, and this will be the actual topic of interest here.

3.2. Cowell's formulation

This method is included in the category of *Special Perturbation Technique* because it deals with the direct numerical integration of the equations of motion, including as many perturbations as the user defines or considers relevant to a particular problem. Precisely the nature of the special techniques is, at the same time, a disadvantage due to the lack of general knowledge they offer[48].

[47] Section 2.1. Subsection 2.1.2.

[48] In this sense, *General Perturbation Techniques* offer the opposite: a better and deeper understanding of the source of perturbation, but they are much more difficult and lengthy as they involve analytic integration of series expansions of the perturbing accelerations (which must be provided analytically, of course).

It is the simplest and most straightforward of all the perturbation methods. It was developed by P.H Cowell in the early 20th century and was used to determine the orbit of the 8th satellite of Jupiter (Callisto) and to predict the return of Halley's comet between 1759 and 1910. It has been "re-discovered" many times in many different forms and it is especially popular and useful nowadays due to the high computational capacity becoming common.

On what concerns to the theoretical basis, the application of this method is basically to write the differential equations of motion of the object under study, to include all the perturbations desired and then to integrate them step-by-step using a brute-force numerical integrator. Thus, it is only a matter of defining adequately Equation (2-19) and carrying out its implementation in the numerical scheme. A common characteristic that all of them exhibit is the need to introduce the system as a linear state-space model by arranging it in first-order differential equations. To do that, the equation of motion is first rewritten as:

$$\ddot{\boldsymbol{r}} + \frac{\mu}{r^3}\boldsymbol{r} = \boldsymbol{a}_p \tag{3-1}$$

Equation (3-1) is the same but with simpler notation. The position and acceleration vectors relative to the dominant body are simply generalised with **r** and μ = G(M+m) ≈ GM (due to the negligible mass of the S/C compared to M).

And then it is arranged as linear state-space constituted by two first-order differential equations:

$$\frac{d\boldsymbol{r}}{dt} = \dot{\boldsymbol{r}} = \boldsymbol{v} \tag{3-2}$$

$$\frac{d^2\boldsymbol{r}}{dt^2} = \dot{\boldsymbol{v}} = \boldsymbol{a}_p - \frac{\mu}{r^3}\boldsymbol{r} \tag{3-3}$$

It must be taken into account that the derivatives are just expressed symbolically for a proper understanding. The reduced system containing two first-order differential equations correspond to the terms on each side of the second equality and are indeed the input of the numerical integrator. In Equations (3-2) and (3.3) **r** and **v** are the position and velocity vector of the S/C relative to the large central body. Finally, the numerical integration is carried out by further broken down into the individual vector components:

$$\frac{dx}{dt} = \dot{x} = v_x \quad ; \quad \frac{d^2x}{dt^2} = \dot{v}_x = a_{px} - \frac{\mu}{r^3}x \tag{3-4}$$

$$\frac{dy}{dt} = \dot{y} = v_y \quad ; \quad \frac{d^2y}{dt^2} = \dot{v}_y = a_{py} - \frac{\mu}{r^3}y \tag{3-5}$$

$$\frac{dz}{dt} = \dot{z} = v_z \quad ; \quad \frac{d^2z}{dt^2} = \dot{v}_z = a_{pz} - \frac{\mu}{r^3}z \tag{3-6}$$

Where:

$$r = \sqrt{x^2 + y^2 + z^2} \tag{3-7}$$

To make this numerical scheme work, initial conditions must be introduced through Equations (2-22) and (2-23). The perturbations alien to gravitational sources shall be added through Equation (2-20) and those arise from the gravity tug of the rest of celestial bodies are introduced through Equations (2-24) and (2-25). Cowell has the advantage of simplicity, but it also has drawbacks:

1) In systems with a large central body, many significant digits must be taken due to the large difference in forces coming from the central and perturbing bodies [18].

2) When motion is near a large attracting body (not necessarily the dominant) smaller integration steps must be taken and round off errors takes its toll [B.4].

3.3. Encke's method

3.3.1. Analytic formulation

Though being a method with more complexity than that of Cowell, this method appeared over half a century earlier, in 1857. Its theoretical foundations differ to that of Cowell in the root, since in this case the sum of all accelerations is no longer integrated altogether. Instead of that, the *difference between the primary and perturbing accelerations is integrated*. The method is based on the concept of *osculating*[49] (or reference) orbit, corresponding to the orbit along which the artificial object would move in the absence of all those perturbing accelerations, thus corresponding to a simple 2BP conic section, where primary body is the dominant (in interplanetary trajectories, usually the Sun).

The true perturbed orbit – which is logically unknown – is in contact or coincides with the osculating orbit at the epoch where the initial conditions apply. Then, the vector analytical expression stating the difference between both acceleration vectors (perturbed and unperturbed) is integrated over time. In case the true orbit deviates too much from the reference orbit, a *rectification* must be made before further integration, consisting on essentially making coincide both orbits at that epoch again (see **Figure 3-1**).

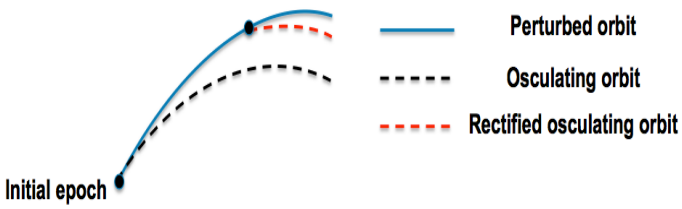

Fig. 3.1 The osculating orbit with rectification.

[49] "*Osculation*" is the "*scientific*" term of "*kissing*". Such term connotes the sense of contact, and, in this case, contact between the reference orbit and the true perturbed orbit (so the name) [B.4]

Deepening now the analytic formulation of the method. Let **r** and ρ be the radius vectors to, respectively, the true (perturbed) and osculating (unperturbed) orbits at a particular time τ (τ = t – t_0). Them the true orbit can be defined as:

$$\ddot{\boldsymbol{r}} + \frac{\mu}{r^3}\boldsymbol{r} = \boldsymbol{a}_p \tag{3-8}$$

And the osculating orbit:

$$\ddot{\boldsymbol{\rho}} + \frac{\mu}{\rho^3}\boldsymbol{\rho} = 0 \tag{3-9}$$

As seen on the picture contained in **Figure 3-2**, at the epoch where the initial conditions are applied, that is, t_0 = 0, it is true that:

$$\boldsymbol{r}(t_0) = \boldsymbol{\rho}(t_0) \quad and \quad \boldsymbol{v}(t_0) = \dot{\boldsymbol{\rho}}(t_0) \tag{3-10}$$

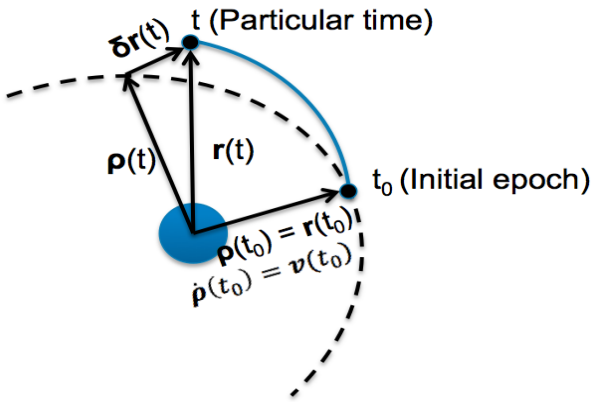

Fig. 3.2 Definition of the analytical deviation vector, δr, from the osculating orbit.

According to the picture, the deviation vector and its second derivative[50] can easily be defined at any particular time as:

$$\boldsymbol{\delta r} = \boldsymbol{r} - \boldsymbol{\rho} \tag{3-11}$$

$$\ddot{\boldsymbol{\delta r}} = \ddot{\boldsymbol{r}} - \ddot{\boldsymbol{\rho}} \tag{3-12}$$

Combing Eqs. (3-8), (3-9), (3-11) and (3-12) and rearranging terms it is obtained the analytical expression wanted that contains the difference between the primary and perturbing acceleration vectors: (written in terms of **r** and **δr**)

$$\ddot{\boldsymbol{\delta r}} = \boldsymbol{a}_p + \frac{\mu}{\rho^3}\left[\left(1 - \frac{\rho^3}{r^3}\right)\boldsymbol{r} - \boldsymbol{\delta r}\right] \tag{3-13}$$

[50] By now, the first derivative of the deviation vector is not necessary.

Equation (3-13) can be integrated over time to get $\boldsymbol{\delta r}$ by simply introducing a set of initial conditions. Given that the osculating orbit is a conic section in an ideal Newtonial gravitational field constituted by two bodies, $\boldsymbol{\rho}$ and its derivative can be defined analytically as a simple function of time instead of numerically integrating[51], so it will be known on each step. Then, Equation (3-11) will allow updating, after each integration step, the position vector of the true perturbed orbit, which was ultimately the objective of the propagation[52].

However, a problem arises in this point: one of the reasons for using this method instead of Cowell was precisely to obtain more accuracy, but the term in parentheses is the *difference of two very nearly quantities*, which requires many extra digits of computer precision on that unique one operation to maintain a reasonable accuracy throughout. A standard method to solve this problem is usually based on defining the following ancillary expression:

$$2q = 1 - \frac{r^2}{\rho^2} \tag{3-14}$$

Thus:

$$\frac{\rho^3}{r^3} = (1 - 2q)^{\ -3/2} \tag{3-15}$$

Introducing Equation (3-15) in (3-13):

$$\ddot{\boldsymbol{\delta r}} = \boldsymbol{a}_p + \frac{\mu}{\rho^3}\left\{\left[1 - (1 - 2q)^{\ -\frac{3}{2}}\right]\boldsymbol{r} - \boldsymbol{\delta r}\right\} \tag{3-16}$$

To compute q, the modulus of the position vector of the perturbed orbit is written in terms of its components, which can be defined in terms of osculating radius and deviation vector components, according to Equation (3-11):

$$\boldsymbol{r^2} = (x^2 + y^2 + z^2) = (\rho_x + \delta x)^2 + \left(\rho_y + \delta y\right)^2 + (\rho_z + \delta z)^2 \tag{3-17}$$

Now expanding the above expression:

$$\begin{aligned} \boldsymbol{r^2} &= {\rho_x}^2 + {\rho_y}^2 + {\rho_z}^2 + \delta x^2 + \delta y^2 + \delta z^2 + 2\rho_x\delta x + 2\rho_y\delta y + 2\rho_z\delta z \\ &= \rho^2 + 2\left[\delta x\left(\rho_x + \frac{1}{2}\delta x\right) + \delta y\left(\rho_y + \frac{1}{2}\delta y\right) + \delta z\left(\rho_z + \frac{1}{2}\delta z\right)\right] \end{aligned} \tag{3-18}$$

Then, dividing at both sides by ρ^2:

$$\frac{r^2}{\rho^2} = 1 + \frac{2}{\rho^2}\left[\delta x\left(\rho_x + \frac{1}{2}\delta x\right) + \delta y\left(\rho_y + \frac{1}{2}\delta y\right) + \delta z\left(\rho_z + \frac{1}{2}\delta z\right)\right] \tag{3-19}$$

[51] This is the best way to proceed, since avoids unnecessary additional numerical truncation errors (which will appear when integrating Equation (3-13) and those ones are unavoidable).

[52] An equation analogous to (3-11) and (3-12) allows obtaining the velocity.

Substituting Equation (3-19) in (3-14) finally an analytical expression for calculating q is obtained:

$$q = -\frac{1}{\rho^2}\left[\delta x\left(\rho_x + \frac{1}{2}\delta x\right) + \delta y\left(\rho_y + \frac{1}{2}\delta y\right) + \delta z\left(\rho_z + \frac{1}{2}\delta z\right)\right] \quad \textbf{(3-20)}$$

With this last expression the system is closed, since it is possible to initialise the process stated in Equation (3-16) from time zero, where $\boldsymbol{\delta r} = 0$ and $\boldsymbol{\rho}$ and its derivative are known from the initial conditions of **r** and **v** as indicated in Equation (3-10). In addition, q = 0 as it can be checked from Equation (3-20). Thus, the first integration would give the first $\boldsymbol{\delta r}$ and its derivative, allowing then calculating **r** and **v** through Equation (3-11) and its analogous, to set the integration for the next step. Then, for the successive steps, $\boldsymbol{\rho}$ and its derivative are perfectly defined for every time, and q can be calculated at the each new step simply applying Equation (3-20)[53].

Now, there is a consistent system to go ahead with the numerical integration that leads to the desired propagation of the perturbed orbit. Nevertheless, despite the variation practiced before in Equation (3-16), the problem is not yet solved since q is as well a very small quantity and seriously compromises the accuracy of the method if no further treatment is done.

Between the several mathematic methodologies for solving the problem at this point, the suggested one consists of performing an expansion of term involving the very small quantities in a binomial series[54] [B.4]:

$$f(q) = 1 - (1-2q)^{-\frac{3}{2}} = 3q - \frac{3\cdot 5}{2!}q^2 + \frac{3\cdot 5\cdot 7}{3!}q^3 - \cdots \quad \textbf{(3-21)}$$

To do so, a short instructions block has been developed to incorporate to the specific function in charge of implementing the method in *MatLab*:

$$\textbf{\textit{Initialisation}}: \quad f_1(q) = 3q \quad ; \quad added_{numerator_1} = f_1(q)$$

$$For\ n = 1\ to\ n\ (desired\ precision)$$

$$\textbf{\textit{Step 1}}: \quad added_{numerator_{n+1}} = added_{numerator_n} \cdot q(3+2n)$$

$$\textbf{\textit{Step 2}}: \quad f_{n+1}(q) = f_n(q) + (-1)^n \frac{added_{numerator_{n+1}}}{(n+1)!}$$

[53] Two things: 1) Shortly, a summary of the computational implementation scheme will be given. 2) Nothing has been commented about the perturbing term $\mathbf{a}_p$, but the way of adding it has already been detailed in *Chapter 2, Section 2.3.2. Ephemeris data management*.

[54] Some time ago, before the advent of the modern computers, an auxiliary function f was defined and tables of f vs q constructed to avoid hand calculation [19]

Some advantages and drawbacks should be noted in Encke's method:

+ This formulation reduces the number of integration steps with respect Cowell's due to the fact that $\boldsymbol{\delta r}$ presumably changes more slowly than **r** does, thus allowing larger step sizes to be taken, which results in:

i) Lesser numerical round off errors.

ii) Generally much less affected by extreme perturbations than Cowell's

- The aforementioned advantages diminish if:

a) The magnitude of the perturbing acceleration becomes much larger than the second term in Equation (3-16) ➔ It means that the osculating parameter (or the orbit itself) needs to be changed since the perturbations are becoming primary.

b) The ratio $\delta r/\rho$ does not remain quite small (typically > 0.01) ➔ It means that a new osculating orbit needs to be chosen, using the concept of rectification[55] (if that event occurs too often, it will indicate that perturbations are affecting more than expected).

To conclude the description of this special perturbation method, the computational algorithm is next outlined. **Appendix B.2** contains the *MatLab* function[56] implemented and those ancillaries for its osculating orbit.

Algorithm operation

1) $Given\ initial\ conditions\ \boldsymbol{r}(t_0) = \boldsymbol{\rho}(t_0), \boldsymbol{v}(t_0) = \dot{\boldsymbol{\rho}}(t_0)\ define\ the\ osculating\ orbit.$
$At\ the\ epoch\ were\ those\ initial\ conditions\ are\ imposed: \boldsymbol{\delta r} = [0\ 0\ 0], \dot{\boldsymbol{\delta r}} = [0\ 0\ 0]$

2) $For\ an\ integration\ step\ \Delta t, calculate\ \boldsymbol{\delta r}(t_0 + \Delta t), knowing\ \boldsymbol{\rho}(t_0), \boldsymbol{r}(t_0), q(t_0) = 0$

3) $Knowing\ \boldsymbol{\delta r}(t_0 + \Delta t), calculate:$
$a)\ \boldsymbol{\rho}(t_0 + \Delta t) \rightarrow Through\ an\ independent\ analytical\ definition\ of\ the\ conic$
$b)\ q(t_0 + \Delta t) \rightarrow Using\ the\ computational\ block\ of\ instructions\ provided\ above$

4) $Integrate\ another\ \Delta t\ to\ get\ \boldsymbol{\delta r}(t_0 + k\Delta t)\ ;\ k\ is\ the\ step\ number\ (k\ is\ synchronised\ with\ the\ adequate\ step\ time, according\ to\ the\ time\ span\ of\ the\ epehemerides\ data)$

5) $If\ the\ ratio \frac{\delta r}{\rho} > a\ rectify\ and\ go\ to\ \boldsymbol{step}\ \mathbf{1}$ ("a" is a specified constant of the user $selection\ but\ of\ the\ order\ of\ magnitude\ of\ 0.001 - 0.01$)

6) $Calculate\ \boldsymbol{r} = \boldsymbol{\rho} + \boldsymbol{\delta r}\ and\ \boldsymbol{v} = \dot{\boldsymbol{\rho}} + \dot{\boldsymbol{\delta r}}$

7) $Go\ to\ \boldsymbol{step}\ \mathbf{3}\ with\ \Delta t\ replaced\ by\ k\Delta t\ (unless\ a\ rectification\ must\ be\ done, hereafter\ only\ \boldsymbol{steps}\ \mathbf{3}\ \boldsymbol{to}\ \mathbf{7}\ need\ to\ be\ applied)$

---

[55] It consists technically on calculating a new osculating orbit making coincide its radius vector and its derivative at that epoch with the radius vector and velocity of the perturbed orbit (therefore, it is equivalent to performing a restoration while preserving the previous states).

[56] It is important to indicate two things: 1) The function is quite generalised, but since was conceived for its application to simulate the interplanetary trajectory BepiColombo, it contains some necessary inputs from the main function. 2) Cowell's method is implemented in two functions: the *static propagator* and the *dynamic propagator* (see Chapter 7). They are basically the same integration of a partial NBP, which considers only the celestial bodies, detected as relevant for all the simulation (static) or only when instantaneously needed (dynamic).

3.3.2. Osculating orbit definition

In the previous subsection, it was stated that ρ and its derivative can be defined *analytically* as a simple function of time instead of numerically integrating, so it can be known on each step. It avoids unnecessary additional numerical errors coming from the truncation. Equation (3-9) corresponds to the two – body equation of motion and the trajectory followed by the S/C relative to the dominant body can be calculated through the *Polar Equation of a conic section*:

$$\rho(t) = \frac{p}{1 + ecos\nu(t)} \tag{3-22}$$

Where[57]:

p = h^2/μ (being h the modulus of the *specific angular momentum*)

e is the *eccentricity* of the conic section

ν is the *true anomaly* (the angle that defines the instantaneous position of the S/C)

As Equation (3-22) shows, the position of the S/C within its conic section (unperturbed orbit) it's a function of time. Actually, if the true anomaly is known for particular times, such position becomes automatically established. Indeed, given that the specific angular momentum – which can be demonstrated, that remains constant along the ideal orbit – is computed through a cross product between the relative position and velocity vectors as follows, nothing impedes to use the initial conditions[58] for calculating it:

$$\boldsymbol{h} = \boldsymbol{\rho} \times \dot{\boldsymbol{\rho}} = \boldsymbol{\rho}(t_0) \times \dot{\boldsymbol{\rho}}(t_0) \tag{3-23}$$

It can be proven as well that, when integrating Equation (3-9) to get the Polar equation, the vector constant arising is actually related to the physics of the problem: it corresponds to the eccentricity vector and its modulus defines the shape of the orbit: *circular* (unusual) *elliptic*, *parabolic* or *hyperbolic*[59].

$$\boldsymbol{e} = \frac{\boldsymbol{\rho} \times \boldsymbol{h}}{\mu} - \frac{\boldsymbol{r}}{r} \tag{3-24}$$

[57] It was decided not to include the steps to integrate Equation (3-9) to get the Polar equation, since it implies performing prior mathematical analyses to prove the existence of the constants of motion: the specific mechanical energy and the specific angular momentum. Given that they are basic and are no longer necessary for the understanding of what remains of this work, it is recommended to deepen them only in case that a lack arises or it is necessary to refresh a particular concept. There are a large number of references to the fundamentals of Astrodynamics and any of them is valid. However, *Chapter 2: Two-Body Orbital Mechanics* from [B.4] is suggested because it is consistent with most of the notation introduced here.

[58] It is worth remembering at this point that the purpose it is to define an osculating orbit for the suitable implementation at each *step 3a* of the Encke's method introduced in the previous subsection. Hence, such initial conditions are those given by the expression (3-10).

[59] It is related with the anomaly problem to be discussed in the next subsection, because the true anomaly can be expressed as a function of angles easier to define, depending on "e" value.

In short, knowing initial conditions, the orbital parameters[60] that define the ideal unperturbed orbit are perfectly determined, with the exception of the true anomaly, which varies over time precisely due to the motion of the object within the orbit. Therefore, what remains to be done in order to obtain the data vector containing all the positions that the S/C would ideally follow for a given osculating orbit, consists in determining the law of time with which the cited true anomaly varies. It depends on the type of orbit and, thus on the eccentricity.

3.3.3. Anomaly problem

The true anomaly that appears in Equation (3-22) can be computed based on auxiliary angles that are easier to define and will depend on the value of the eccentricity. These are *eccentric* (0<e<1), *parabolic (e=1)* and *hyperbolic (e>1)* anomalies. The theoretical development will be omitted again as it can be easily found in *Chapter 4: Position and velocity as a function of time* in Bate's book [B.4] and does not provide any additional knowledge that may be necessary hereafter for the development (in fact, an interplanetary trajectory for the most part follows a perturbed elliptical conic).

First, a guideline will be introduced that contains the common steps to follow regardless of which of these three cases of eccentricity applies, since a function implemented in *MatLab* will be in charge of performing the calculations each time it is necessary to define a new osculating orbit (i.e., at initialisation or at rectification, if needed):

1) The classical orbital elements are computed from the initial conditions given by (3-10). Actually, the rest of them, since "e" was stated in Equation (3-24).

2) Given that the orbital elements remain constant in an unpertubed orbit, the mean anomaly is then computed in the current time t and *Kepler's equation* is then solved to obtain the corresponding anomaly (eccentric, parabolic or hyperbolic). Except for the parabolic case, *Newton – Raphson* will be applied.

3) Once the anomaly corresponding to the type of orbit is known in t, it is possible to calculate the true anomaly and then the modulus of the radius vector stated in Equation (3-22). Finally, since that is only the modulus, both the radius vector and its derivative (the velocity vector) will be computed first in perifocal coordinates and then transformed in the proper central[61] reference frame (i.e, the Planeto-centric-Equatorial when the S/C orbits a planet and the Heliocentric-Ecliptic when it does about the Sun).

---

[60] In fact, the following subsection will verify that this is true. The initial conditions allow computing all the orbital elements necessary to "*draw*" and orient the orbit in a three–dimensional space (i.e., a, e, i, Ω and ω) the remaining (ν) being the only one necessary to locate the instantaneous position of the object on it.

[61] It should not be forgotten that the initial conditions are obtained with from Equations (2-22) and (2-23) conveniently expressed with respect to the corresponding dominant body. Hence, there is consistency and no need for additional arrangements between the reference systems when computing the osculating orbit. Moreover, the formulation to be introduced to transform from perifocal to the central dominant body is of universal application, since it is based on orbital elements that were obtained precisely by using a state vector relative to the dominant body.

Applying the first step described in the above guideline, the remaining orbital elements shall be calculated. To determine the *semi-major axis*, simply:

$$a = \frac{h^2}{\mu(1-e^2)} \tag{3-25}$$

The vector pointing to the ascending node is needed. It can be computed using geometric relationships of the conic section that is defined by its orbital elements. **Figure 3-3** will help to spatially visualise the vectors involved.

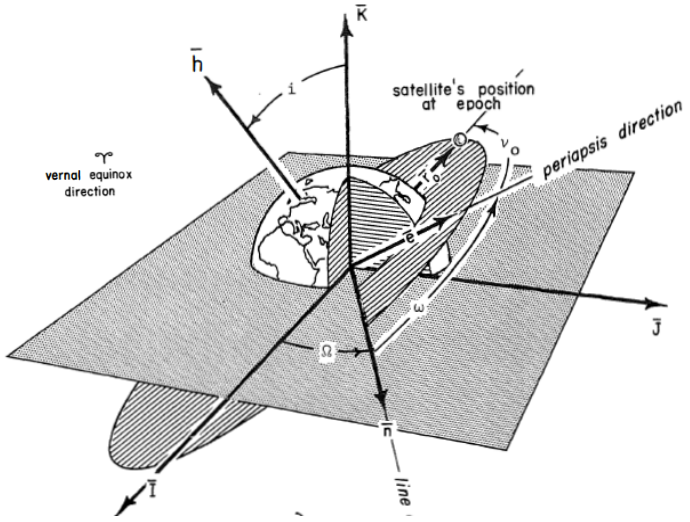

Fig. 3.3 Orbital elements that univocally define an orbit in the space (the picture shows the origin of frame in the Earth, but it is valid for any major body).

$$\boldsymbol{n} = \boldsymbol{K} \times \boldsymbol{h} \tag{3-26}$$

Next, the *inclination* of the orbit, which is always less than π rad:

$$i = \cos^{-1}\frac{h_z}{h} \tag{3-27}$$

The longitude of the *ascending node* will depend on the previous direction obtained for the *node vector* since it is the angle that it forms with the X-axis:

$$\Omega = \begin{cases} \cos^{-1}\dfrac{n_x}{n} & if\ n_y \geq 0 \\ 2\pi - \cos^{-1}\dfrac{n_x}{n} & if\ n_y < 0 \end{cases} \tag{3-28}$$

The *argument of periapsis* will depend on the previous direction obtained for the *node vector* as well since it is the angle that it forms with the absides line:

$$\omega = \begin{cases} \cos^{-1}\dfrac{\boldsymbol{n}\cdot\boldsymbol{e}}{n\cdot e} & if\ e_z \geq 0 \\ 2\pi - \cos^{-1}\dfrac{\boldsymbol{n}\cdot\boldsymbol{e}}{n\cdot e} & if\ e_z < 0 \end{cases} \tag{3-29}$$

And finally, the *true anomaly* can be calculated as the angle formed between the eccentricity vector pointing to the line of absides and the current S/C position vector, which is known because is of the initial conditions gathered from (3-10):

$$\nu = \begin{cases} \cos^{-1}\dfrac{\boldsymbol{e}\cdot\boldsymbol{\rho}(t_0)}{e\cdot\rho(t_0)} & if\ \boldsymbol{\rho}(t_0)\cdot\boldsymbol{\rho}(\dot{t}_0) \geq 0 \\ 2\pi - \cos^{-1}\dfrac{\boldsymbol{e}\cdot\boldsymbol{\rho}(t_0)}{e\cdot\rho(t_0)} & if\ \boldsymbol{\rho}(t_0)\cdot\boldsymbol{\rho}(\dot{t}_0) < 0 \end{cases} \quad \textbf{(3-30)}$$

Once applied all these previous equations, the position of the S/C relative to the dominant body within the unperturbed orbit, at the epoch when the initial conditions are provided by the ephemeris, is completely determined[62]. This is the end of the *step 1* contained in the guidelines introduced above.

Entering now in the next block of instructions, before going ahead it is important to properly understand what is going on. Currently, barely the first state that S/C exhibits within the unperturbed orbit has been defined, which is represented by Equation (3-9), i.e, the state corresponding to the initial conditions, which is the same for both the unperturbed and perturbed orbits. Nevertheless, the core of the analytic construction of the conic section is already established and there will be no need of change it anymore, unless a rectification is required at some point within the Encke's execution (which does not affect the previous expressions, having been properly implemented within the algorithm).

Therefore, on what consists this second step is to develop the expressions that will allow using the initial true anomaly calculated by Equation (3-30) to perform a recursive scheme for getting those corresponding for every time. Despite it was stated that the three steps are common for each case, let is calculate the second particularised for each case as to see the differences.

Elliptic Case

Allowing recursive calculations until getting the number of elements that $\rho(t)$ must have as to analytically define the whole osculating orbit (imposed by the step size of the integration, which in turn is synchronised with the time span of the ephemeris data) each updated value for the true anomaly is to be calculated by solving iteratively Kepler's equation for each loop step.

First, the *eccentric anomaly* is computed as follows:

$$E = 2\tan^{-1}\frac{\tan\frac{v}{2}}{\sqrt{\frac{1+e}{1-e}}} \quad \textbf{(3-31)}$$

[62] It is just a matter of substituting the last result of Eq. (3-30) in Eq.(3-22).

Next, the *mean anomaly* can be calculated directly from Kepler's equation:

$$M = E - e \sin E \tag{3-32}$$

Once every classical orbital element is known, it is the moment to update the new dynamic state (for t_1). To do so, the first to be calculated is the *mean anomaly* itself[63]:

$$M_{t1} = M_{t0} + \sqrt{\frac{\mu}{a^3}}(t_1 - t_0) \tag{3-33}$$

Now, Kepler's equation needs to be solved using a numerical method to iteratively solve no linear equations. As indicated, it is carried out through *Newton – Raphson*:

$$f(E) = E - e \sin E - M \tag{3-34}$$

$$E_{i+1} = E_i - \frac{f(E_i)}{\frac{d}{dE_i} f(E_i)} \quad ; \quad E_0 = M$$

Once solved the matter of getting an updated eccentric anomaly, it is possible to ultimately computing the pursued true anomaly. It is accomplished using the same Equation (3-31) and isolating but it will be introduced for the sake of clarity:

$$v(t) = 2 \tan^{-1}\left(\sqrt{\frac{1+e}{1-e}} \tan \frac{E(t)}{2}\right) \tag{3-35}$$

An emphasis has been conferred by explicitly introducing the time in the notation to leave clear that Eqs. (3-31) to (3-35) are introduced within a loop as to recursively carry out the calculations to obtain all the elements of ρ (t). For its implementation in *MatLab*, it worth instead of performing comparisons between adequate quadrants – by incorporating the corresponding block of conditional statements – using the command "*atan2*" instead of the regular "*atan*".

According to *MatLab*, such function is defined as:

[63] Two things: 1) M_{t0} of Equation (3-33) is exactly the M of Equation (3-32). A change of notation has been introduced (and avoided previously to not saturate the notation itself).

2) Remind that t_1 = 6 h, t_0 = 0 in our particular case.

$$atan2(y,x) = \begin{cases} \tan^{-1}\dfrac{y}{x} & if\ x > 0 \\ \tan^{-1}\dfrac{y}{x} + \pi & if\ y \geq 0, x < 0 \\ \tan^{-1}\dfrac{y}{x} - \pi & if\ y < 0, x < 0 \\ \dfrac{\pi}{2} & if\ y > 0, x = 0 \\ -\dfrac{\pi}{2} & if\ y < 0, x = 0 \\ \text{undefined} & if\ y = 0, x = 0 \end{cases} \quad \textbf{(3-36)}$$

In consequence:

$$v(t) = 2\, atan2\left(\sqrt{1+e}\sin\frac{E(t)}{2}, \sqrt{1-e}\cos\frac{E(t)}{2}\right) \quad \textbf{(3-37)}$$

Parabolic Case

The initial *Parabollic eccentric anomaly* is obtained in a similar fashion to the true anomaly of the previous case:

$$D = \tan\frac{v}{2} \quad \textbf{(3-38)}$$

Next, the mean anomaly:

$$M = 2\sqrt{\frac{\mu}{p^3}}\left[D + \frac{1}{3}D^3\right] \quad \textbf{(3-39)}$$

By carrying out a similar update of new dynamic state (for t_1) as before:

$$M_{t1} = M_{t0} + 2\sqrt{\frac{\mu}{p^3}}(t_1 - t_0) \quad \textbf{(3-40)}$$

Now, taking into account how the mean anomalies are defined for the parabolic case [B.4]:

$$M_{t1} - M_{t0} = 2\sqrt{\frac{\mu}{p^3}}\left[\left(D_{t1} + \frac{1}{3}{D_{t1}}^3\right) - \left(D_{t0} + \frac{1}{3}{D_{t0}}^3\right)\right] \quad \textbf{(3-41)}$$

The inverse problem (what is needed to isolate the updated value of the parabolic eccentric anomaly) can also be solved applying Newton – Raphson.

$$f(D) = 2\sqrt{\frac{\mu}{p^3}}\left[D + \frac{1}{3}D^3\right] - M \quad \textbf{(3-42)}$$

$$D_{i+1} = D_i - \frac{f(D_i)}{\frac{d}{dD_i} f(D_i)} \quad ; \quad D_0 = M$$

Finally, analogously (using the command "*atan2*" as before):

$$\upsilon(t) = 2\tan^{-1} D \tag{3-43}$$

Hyperbollic Case

The initial *hyperbolic eccentric anomaly* can be obtained directly using an expression that relates it to the true anomaly:

$$F = 2\tanh^{-1}\left(\frac{\tan\frac{\upsilon}{2}}{\sqrt{\frac{e+1}{e-1}}}\right) \tag{3-44}$$

Now the mean anomaly is obtained in a similar fashion from Kepler's equation applied to the hyperbolic case:

$$M = e\sin F - F \tag{3-45}$$

By carrying out a similar update of new dynamic state (for t_1) as before:

$$M_{t1} = M_{t0} + \sqrt{\frac{-\mu}{a^3}}(t_1 - t_0) \tag{3-46}$$

The inverse problem can be solved applying Newton – Raphson again:

$$f(F) = e\sin F - F - M \tag{3-47}$$

$$F_{i+1} = F_i - \frac{f(F_i)}{\frac{d}{dF_i} f(F_i)} \quad ; \quad F_0 = M$$

And finally, analogously (using the command "*atan2*" as before):

$$\upsilon(t) = 2\,atan2\left(\sqrt{e+1}\sinh\frac{F(t)}{2}, \sqrt{e-1}\cosh\frac{F(t)}{2}\right) \tag{3-48}$$

Once applied the second step of the guideline for each particular case, all what remains is just the transformation stated at the third step in order to get $\rho(t)$ and its derivative in body-centred instead of in perifocal. To do so, it is only necessary to make use of the five constant[64] orbital parameters (a, e, i, Ω and ω) and the time-variable $\nu(t)$.

[64] As indicated earlier, they remain constant for a given osculating orbit. If a rectification takes place when applying Encke's method, the algorithm will need to re-compute them.

$$\begin{pmatrix}\rho_I\\ \rho_J\\ \rho_K\end{pmatrix}=\begin{pmatrix}R_{11} & R_{12} & R_{13}\\ R_{21} & R_{22} & R_{23}\\ R_{31} & R_{32} & R_{33}\end{pmatrix}\begin{pmatrix}\rho(t)\cos\ \upsilon(t)\\ \rho(t)\sin\ \upsilon(t)\\ 0\end{pmatrix} \quad \textbf{(3-49)}$$

$$\begin{pmatrix}\dot{\rho}_I\\ \dot{\rho}_J\\ \dot{\rho_K}\end{pmatrix}=\begin{pmatrix}R_{11} & R_{12} & R_{13}\\ R_{21} & R_{22} & R_{23}\\ R_{31} & R_{32} & R_{33}\end{pmatrix}\sqrt{\frac{\mu}{p}}\begin{pmatrix}-\sin\upsilon(t)\\ e+\cos\upsilon(t)\\ 0\end{pmatrix} \quad \textbf{(3-50)}$$

Where **I**, **J** and **K** are the components in the body-centred coordinate system (planeto-centric or Helio-centric, depending who the S/C is orbiting about). During cruise phase, the dominant body is the Sun, at the departure the Earth and, at each flyby, the corresponding planet (remember that, to avoid problems, it was mentioned in subsection *2.3.2. Ephemeris data management* that, the propagated state vectors must be expressed with respect SSB again after each integration step, to ensure they remain in the original form from where formulated when they are not being integrated). And the *rotation matrix*:

$$\boldsymbol{R_{11}}=\cos\Omega\cos\omega-\sin\Omega\sin\omega\cos i\quad \boldsymbol{R_{12}}=-\cos\Omega\sin\omega-\sin\Omega\cos\omega\cos i\quad \boldsymbol{R_{13}}=sin\,\Omega\,sin\,i \quad \textbf{(3-51)}$$

$$\boldsymbol{R_{21}}=sin\,\Omega\,cos\,\omega+cos\,\Omega\,sin\,\omega\,cos\,i\quad \boldsymbol{R_{22}}=-sin\,\Omega\,sin\,\omega+cos\,\Omega\,cos\,\omega\,cos\,i\quad \boldsymbol{R_{23}}=-\,cos\,\Omega\sin i \quad \textbf{(3-52)}$$

$$\boldsymbol{R_{31}}=sin\,\omega\,sin\,i\quad \boldsymbol{R_{32}}=cos\,\omega\,sin\,i\quad \boldsymbol{R_{33}}=cos\,i \quad \textbf{(3-53)}$$

CHAPTER 4. TOOL DEVELOPMENT STRATEGY

At this point, all the necessary background and motivation are already on the table (*Chapter 1*) along with the technical content associated with physics and mathematics (*Chapters 2 and 3*) necessary to **build**, **apply** and **validate** the tool/procedure that will lead to the desired analyses and conclusions.

This is a brief chapter that essentially seeks the construction of a logical process that will lead to the achievement of the objectives set at the beginning of this thesis. It will start from a reminder of exactly what being pursued, taking motivation section by the hand and limiting itself to presenting its central ideas summarised. Next, different sub-sections will be entrusted to the description of the strategies developed to achieve each specific objective necessary to carry out the next step. It will be done avoiding deepening the implementations and results, as will be extensively described in later chapters. Such a description will try to follow a logical and natural order, especially in accordance with the chronology of the ideas as they were conceived, analysed and finally consolidated to be carried out. It will be very important that each development makes it possible to understand why a step is taken in a one direction and not in another. Finally, a roadmap will be included in a schematic flowchart.

Before entering each sub-section, let us remember the cornerstones associated with the present work:

- Relationship to the JPL ephemeris model, which has the most accurate description of motion in the Solar System. It provides high-precision ephemerides in support of spacecraft navigation and other activities, by integrating the full dynamical NBP (i.e, including every relevant perturbation as *thruster firings*, *SRP*, *extended spherical harmonic gravity fields*, *drag*, and whatever other relevant dynamic model is used)

- Interest in deepening the understanding of the influence of perturbations on the trajectory calculations made through this model, particularly by studying those of a gravitational nature in *interplanetary trajectories* due to the following (by priority):

  1) It leads to large requirements and machine–oriented technical resources spent in terms of both time and computational cost (very long missions).

  2) During its cruise phase (longest leg) the S/C is affected by a multi-body attraction, rather than by a reduced number of celestial bodies and limited effects.[65]

[65] E.g, a satellite orbiting the Earth can be accurately modelled by the RTBP, taking into account the Sun, the S/C and the Earth itself, as it moves close enough to the planet to be perturbed essentially by the Sun and not by many other planets. In this case, the model plus external disturbances such as drag/SRP (one of both depending on the orbit altitude) and the J_2 term of Earth's non-spherical gravity potential is sufficient to propagate a very acceptable trajectory.

3) Whether a computational brute force or a more sophisticated technique is used, the algorithm will require continuous updating of the dynamic states (positions and velocities) of a large number of bodies. Thus, here is the greatest potential for time saving if any tool can identify, for a given interplanetary trajectory, which celestial bodies do not exert a significant difference in the motion of the S/C and can thus be neglected without loss of accuracy (the savings would come from avoiding the access and download from the platform of many planetary ephemerides, as well as avoiding long loading times of large data array sets in the workspace of the programming environment).

4) By referring to the term "loss of accuracy" the concept of "comparison" implicitly arises. Indeed, a comparison exists, but the purpose of the thesis is not to refine simpler models and compare them with the JPL ephemeris (other works focus on it) but to make such a comparison between the application of the full NBP and that of the partial NBP (i.e., including only in the perturbing term the celestial bodies declared relevant by the developed tool)[66].

5) The criteria for defining when a perturbation is relevant or not are given in terms of the ΔV needed to perform correction manoeuvres at intermediate points of the path, rather than in terms of distance. The reason is that in terms of distance it is complicated to know what is too much or too little in the framework of the complex multi-body gravitational field. Therefore, if ΔV_{corr} implies deviations greater than the discrepancy when comparing the full NBP and partial NBP, said discrepancy will be considered not significant and, consequently, the exclusion of those removed bodies without loss of accuracy will be accepted.

4.1. Concept of perturbative acceleration threshold

Let us imagine an interplanetary trajectory with the goal of intercepting a given asteroid. Once the spacecraft leaves the parking orbit from Earth and is injected into the transfer orbit – which can be achieved through the launcher itself – any necessary re-targeting or correction manoeuvre must be carried out entirely by the main propulsion system on-board. **Figure 4-1** shows a rendezvous trajectory for a given asteroid for particular dates (departure/arrival) achieved through the application of a *Lambert Arc solver*.

If only the Sun and nothing else composed the infinite space, the Two-body problem would mathematically lead to a perfect prediction of those rendezvous trajectories, since Lambert Arc solver allows a minimum energy transfer orbit to be fitted[67] to a particular time-of-flight (defined by the two departure/arrival dates). Therefore, such a simulated trajectory would be exactly the realistic one performed by the spacecraft once it leaves the Earth.

---

[66] The greatest savings will not come from neglecting perturbations other than those gravitational (which, moreover, are mostly introduced directly into the perturbing term as analytical expressions, without the need to configure large data matrices).

[67] The solver incorporates an internal differential corrector and a continuation method to perform a re-targeting of the velocity vector at the departure the necessary number of times until the desired tolerance is reached.

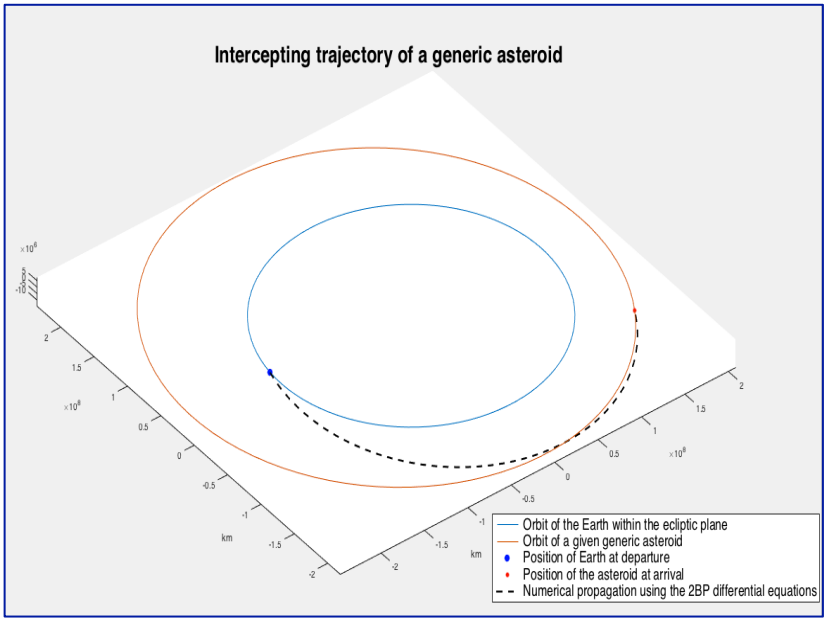

Fig. 4.1 Asteroid rendezvous trajectory using a Lambert Arc solver (*Two-body Orbital Boundary-value problem*).

Unfortunately, nature is never so easy and there are planets and other celestial bodies that perturb the motion around the Sun, as they also have their own gravitational fields. Therefore, in order to minimise deviations[68], an integration of the NBP general equations of motion should be carried out rather than using 2BP. The question that arises is: *What bodies should we consider to achieve sufficient precision? (i.e., a small deviation)*

Obviously as the name of "N-bodies" itself expresses, the greater the number of bodies considered, the greater the accuracy in the computed trajectory, since it means approaching the real natural behaviour. However, it requires high computational costs, which again leads to the question raised earlier. The answer is inevitably related to the ***propulsive characteristics of the S/C*** because, depending on how much acceleration it can develop, the influence of the perturbing effects will vary.

Asteroid rendezvous may involve crossing areas of space where the vector sum of the perturbing accelerations can theoretically be cancelled using the main spacecraft engine – or attitude thrusters if it is small enough. Nevertheless, when its magnitude exceeds a certain **threshold** – let us say, the maximum acceleration capacity of the propulsion plant – it will not be possible to cancel the disturbances and thus they must be considered.

[68] One might think that the trajectory shown in **Figure 4-1** has not deviations or, if it does, they are very small. What happens is that Lambert Arc perfectly solves a problem in which the underlying physics involves only two masses (that of an object moving about a dominant central body). In other words, neither the planet from which the departure occurs nor the planet to which the arrival occurs are taken into account as masses at any time. Rather, they are only considered as locations to establish a couple of dates: In which the S/C has to depart and the one that has to arrive (to fit mathematically and univocally into the trajectory). Since real physics differs from such portrait, technically the Lambert Arc solves a "fictitious problem".

Figure 4-2 shows both extreme cases for get an intuitive idea. It represents the *Sun*, the *Earth* and the *asteroid* (brown) at two different times along the rendezvous trajectory, symbolised by dashed lines. From one hand, the blue vector corresponds to the *first term* of Equation (2-19) and points to the centre of the Sun as corresponds to the direct acceleration exerted on the spacecraft. On the other hand, the black vector corresponds to the *second term* (i.e., the perturbing term in the equations of motion) that points in a certain direction at that time, according to the instantaneous vector sum[69] of the perturbing accelerations coming from the Earth and the rest of celestial bodies (not shown here). The green vector corresponds to the acceleration produced by the S/C – when it is enough to cancel the perturbations – or in red in the opposite case.

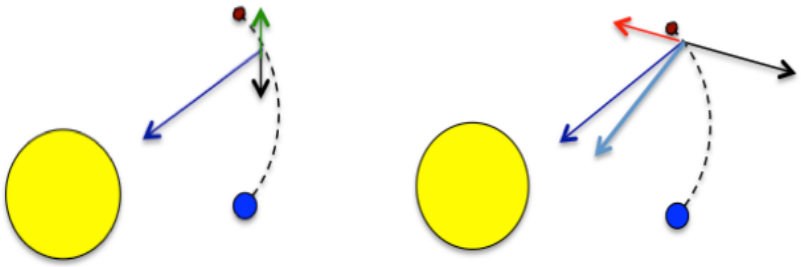

Fig. 4.2 Asteroid rendezvous trajectory showing positions of the S/C at two different times: Left) the S/C cancels the perturbing effects. Right) the S/C does not have enough acceleration capability to cancel the perturbing effects.

In the picture on the left, there are only three vectors drawn because the S/C perfectly cancels the vector sum of the perturbing acceleration, so the resultant points towards the centre of the Sun. Nevertheless, at another time (picture on the right) the magnitude of the vector sum of the perturbing acceleration is greater than the maximum achievable by the S/C and cannot be completely cancelled[70], thus appearing a fourth vector (soft blue) since now the total acceleration exhibited by the S/C relative to the Sun does not point towards its centre. At that very moment, if the celestial bodies that caused such perturbance are not considered in the general equations of motion, the trajectory will suffer a disagreement with that of 2BP, due to the deviation produced (in fact, the S/C will not reach the desired target on the imposed date)

From the above imaginative and qualitative analysis, the conclusion is clear: a *threshold must be chosen* to differentiate when a perturbation exerted on the moving S/C should be considered relevant and when not. At this point, it is crucial to point out that, if this simple but necessary reflection were not performed, a tool or procedure leading to such a determination would not be possible to implement. An alternative to study the relevance of perturbations could be directed by large-scale statistic analysis or by Monte Carlo simulations,

[69] An attempt has been made to emphasize the direction pointing closer to the Earth when the S/C is at a point within its trajectory that is closer to it (left) compared to what it does later (right).

[70] By the time being, let us call it real propulsion capacity. In the next subsection it will be discussed that actually ESA's margin philosophy guideline suggests basing the threshold in terms of a certain fraction of these real propulsion capabilities.

among others, probably not being the best options[71] given the dynamic, chaotic and complex nature of such vector system of non-linear second order differential equations.

Apart from what has been said, a threshold based on the acceleration capabilities of a spacecraft encourages the tool/procedure to be applicable only to missions designed with that same space vehicle, resulting in an inevitable lack of generality. Nevertheless, this can be offset by *two facts*:

1) If the threshold is based on the propulsive characteristics of a modern spacecraft (i.e., representative of today's times) the tool developed could be applied to any type of trajectory of a planned interplanetary mission, thus exhibiting a small "oasis" of generality.

2) Although many different types of spacecraft are conceived for various interplanetary missions[72] (depending on their scientific purposes and the nature of the designed trajectory) and the tool developed for a particular S/C was not universally applicable, the *procedure* could be reproduced as many times as desired, regardless of the particular case (i.e, reproduce the logical steps to first establish a threshold and then implement the tool to detect the relevant planetary disturbances).

4.1.1. Rationale to take as reference a low-thrust propulsion plant

Having introduced the concept of threshold involving discrimination between when the vector sum of perturbative acceleration is relevant or not for a given interplanetary trajectory, now is time to base it on a suitable representative.

The first rough estimate was initially based on information obtained from the *Mars Express* mission [20, 21], which was a successful mission performed using a small spacecraft. Its main propulsion system was used solely for orbit corrections and to slow the spacecraft down for Mars orbit insertion, unlike a larger one such as *Cassini*, for instance. The reason for putting a reference point on a small spacecraft is that it is a good representative of current missions. On the other hand, a S/C has a negligible mass compared to celestial bodies; so there will be not significant variation in the level of gravitational attraction as a function of its dimensions. Although the relationship between mass, size and propulsion is by no means linear[73], smaller spacecraft tend to exhibit small (low-power) propulsion plants and this is what ultimately

[71] By pinpointing this does not mean to be alluding that they are not interesting studies (in fact they are and probably many of them have already been carried out). What is meant is that they are not the best option if a deterministic tool or procedure is to be developed, as is the case.

[72] As will be widely detailed in ***Chapter 5****. Estimation of the perturbing acceleration threshold*, there was (and likely still is) a cast of spacecraft with very different characteristics – in terms of propulsion, dimensions and mass – that fitted to different types of interplanetary missions.

[73] This relationship will be object of study in the next Chapter and the non-linearity duly demonstrated.

determines[74] which perturbations affect the S/C.

Going a step further, it will not even be the size of the propulsion plant that defines whether the power is "small" or "large" (i.e., the acceleration capacity) but the nature of the propulsion itself on which that plant is based (i.e, chemical, electric, nuclear, magnetic…etc). Therefore, a ***worst-case scenario*** could be based on data extracted from the *lowest powered spacecraft*[75] possible to calculate the perturbing acceleration threshold.

In this sense, the *Mars Express* mission will not be a good reference point, since the inconvenient of basing a threshold on chemical plants is that it corresponds to high-thrust (large acceleration capabilities) which means that whatever the interplanetary mission is, the significant gravitational bodies will always be those of which the S/C departs from (e.g., Earth) those that are reached or approximated (asteroid or planet) and the Sun (as perturbation or as dominant body, according to the different phases)[76].

To solve this drawback, the reference will be taken from an interplanetary trajectory developed using a low-thrust electric propulsion plant. The starting point will be the *SMART-1* mission, which was the first to use electric propulsion both to escape from the Earth's gravitational field and to inject itself into an interplanetary transfer orbit [22]. The analysis leading to the final establishment of the threshold – which will require two iterations –will be carried out entirely in *Chapter 5*.

Before moving on to the next section to further develop the concepts to build the tool/procedure pursued, let is briefly summarising some of the reasons why it has been decided to base the data to calculate the perturbative acceleration threshold in past electrically propelled low-thrust missions:

- The lower the order of magnitude of the threshold, the worse the scenario will be considered and, therefore, more zones of space will exert a perturbation on the spacecraft greater than that value (for being pessimistic or conservative)
- A greater sensitivity to perturbing accelerations will allow the identification of the most significant bodies when deviations from a desired trajectory occur
- It could be representative of the real perturbing environment that a small interplanetary spacecraft has to deal with when it has to intercept an asteroid once departed from Earth

[74] It is obvious that actually the influence of such variations in mass and size will contribute to a greater or lesser extent, as it corresponds to the state-of-the-art, which is continuously being improved technologically (thus, it changes).

[75] In any case, without forgetting that it should be a representative of modern spacecraft today.

[76] Note that it would only involve three bodies at any one time: 1) At the departure, the dominant body is the Earth and the only relevant perturbation comes from the Sun. 2) During the interplanetary trajectory, there are not significant perturbations due to the high-thrust capabilities exhibited by the S/C. 3) At the arrival, the dominant body is the asteroid/planet/natural satellite and the only relevant perturbation comes solely from the Sun again.

4.2. Sun gravity field perturbed by planets

Although what has been introduced so far hardly corresponds to the "first stone", as is often the case in the world of science, the concept itself is the cornerstone for solving a problem, while the rest is simple "mathematical masonry".

Continuing with the exposition of the strategy for the construction of the tool, the next step will be to use the general equations of motion obtained in Equation (2-19) but instead of integrating them (which will be done later to validate them) they will be evaluated directly in a control volume[77] that will contain a discretisation of positions of the three-dimensional space that surrounds the Earth and that will represent the part of the Solar System under study. The information obtained will be represented by contourplots showing the magnitude of the perturbing acceleration in the solar gravitational field. Since the parameter that defines the boundary of such contourplots is the threshold computed previously, they will show the areas[78] within the control volume where the S/C is subject to significant perturbations (i.e, those exceeding its threshold; they will be shown by the colour that the graph defines to establish a boundary).

4.2.1. Mapping of the Solar System under study

In this subsection, a mapping of a part of the Solar system will be done, first defining a suitable control volume surrounding the Earth and then creating a *2D spatial discretisation based on a planar projection approach*.

First, let us to delve into the concept of "*planar projection*", which involves the transfer of information contained in a three-dimensional region to a single of only two dimensions. So, the first question that comes up is, *why to do that instead of focussing directly on three-dimensional space*?

Figure 4-3 will help to understand the idea. It shows a series of arrows departing from the centre of a planet (i.e., the Earth). Each of them points in a certain direction and all their ends are at the same distance. It has been obtained by discretising a spherical layer using spherical coordinates.

[77] The reason for evaluating the general equations of motion is motivated by the interest in studying the volume of part of space, where theoretically the S/C can move within a given interplanetary trajectory. In other words, it is interesting to know how space would "react", talking in perturbing terms, to the presence of the S/C in each position.

[78] These areas or zones will usually be referred to as *Perturbation Zones* or simply *PZ*.

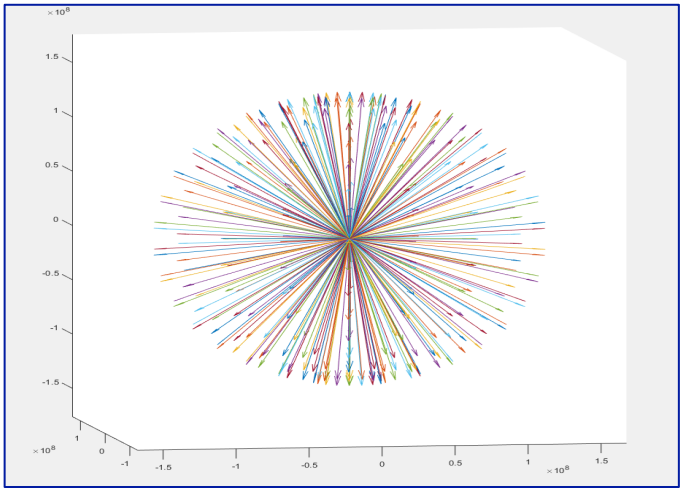

Fig. 4.3 Spherical discretisation of positions around a central body.

The problem associated when attempted to discretise space following this concept is enormous. First, multiple spherical layers must be generated in order to cover a sufficiently large and fine space region. Second, if a representative region is desired, it is necessary to establish a very small angular separation with respect to the z-axis (±180°) and with respect to the x-axis[79] (±360°). Such a discretisation scheme – carried out in *Matlab* – involved three nested "*for*" loops, which inevitably leads to high times and computational costs. Taking into account the distance scales in space, performing a spatial discretisation with sufficiently fine positions grid, would be, at a minimum, very *inefficient* from a computational point of view.

A next step – and a suggestion agreed with the supervisor – was then to try to make a planar approach. **Figure 4-4** corresponds to a XZ view (perpendicular to the ecliptic plane) and shows the trajectory that the asteroid IRIS will describe for a time equal to its sidereal period, along with the Sun and planets from Mercury to Jupiter. The dynamic states (in this case only positions) were represented using ephemeris gathered from the HORIZONS platform. Due to the slowness of the orbits of the planets beyond Jupiter, it was decided not to include them in the graph, since for the simulated time and due to the scale of distances, it makes the interpretation of the picture more difficult.

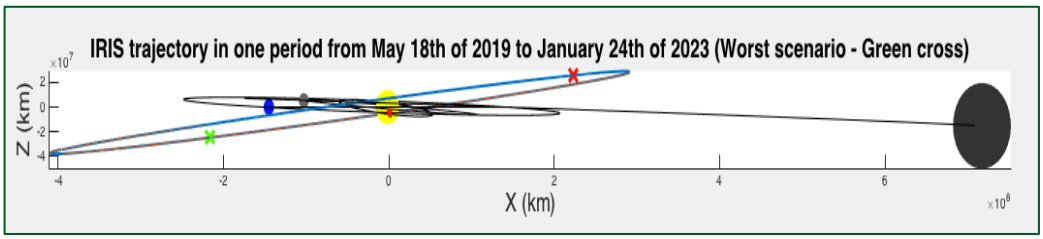

Fig. 4.4 IRIS orbit and some planets of the Solar System. Ephemeris represented for a time corresponding to the asteroid sidereal period.

[79] In the case of the X-axis, it refers to the projection of each arrow in the XY plane.

As it can be seen, the Solar System exhibits by its very nature, a quite acceptable "*flatness*". Even asteroids or comets that describe orbits farther from the ecliptic plane (extreme cases) do not exhibit such outlandish inclinations. Since, in addition, in the present case the focus regarding perturbations, is not centred in asteroids but on planets (Mercury exhibits the orbit with greatest inclination with 7°) the planar projection[80] was finally *accepted*[81] *as representative of the region of the space under interest* (i.e, the control volume).

The last question that remains unanswered is what the extent of the control volume will be. Even in 2D, execution times are high if large and very fine extensions are generated. On the other hand, a discretisation is a grid of positions and, the finer it is, the better the reality described (positions closer to each other). Thus, it is a trade-off. Some preliminary analyses showed that, with the threshold set (see next chapter) the Perturbation Zones[82] of outer planets as *Jupiter* and *Saturn*, were not reaching the orbit of Mars (although they did not stay too far away). Since it is a question of achieving a compromise between the description of reality and computational resources, and the most interesting applications focus on Earth's surroundings, it was finally decided to *extend the control volume to slightly beyond Saturn's orbit*[83].

It is important to clarify that, in the end, perturbations come from the fact that each planet gravitationally distorts the space around it, so it could happen that either these perturbative spheres are isolated, or that they intersect or even that the perturbations of the massive outer planets "engulf" the orbits of the interiors. As this does not happen, the volume of control taken was considered acceptable and representative for Earth's interplanetary missions to other internal planets (or to other external planets without surpassing Saturn's orbit).

On the other hand, the development of a tool containing a discretisation covering up to Saturn does not mean that the subsequent validation (**Chapter 7**) neglects the remaining planets. Indeed, all of them will be included to carry out the propagation of trajectories and the discrepancy analyses. What it really means is that, later mission trajectory designs making use of the capabilities of the tool should be restricted to the control volume.[84] All the technical details will be discussed in the **Chapter 6**. *Maps of perturbed Sun gravitational field*.

---

[80] Its construction will be introduced in ***Chapter 6****. Maps of perturbed Sun gravitational field*.

[81] In **Chapter 7**, the validation process will demonstrate that it is a very good approach.

[82] A reminder: PZs are the areas within the control volume in which the S/C is subject to a magnitude of the vector sum of perturbing acceleration exceeding its threshold.

[83] It should be noted that the definition of the control volume is reasonable, since if Jupiter's PZ does not even reach the Martian orbit, those of Uranus, Neptune and Pluto (taking into consideration the scale of distances) would probably not even reach their neighbourhood.

[84] As mentioned earlier, the *procedure* could be reproduced as many times as desired, regardless of the particular case. Hence, if the mission requirements include traveling beyond Saturn, the 2D maps can be made obtained by following the same routine and implementing the algorithms with a larger (and finer if apply) grid.

4.2.2. Evaluation of motion equations in spatial discretisation

Once obtained the spatial discretisation of the control volume, what we have is essentially a grid of positions where we can evaluate Equation (2-19) to obtain the magnitude of the vector sum of the perturbing accelerations. The evaluation could be carried out following two approaches:

1) **Dynamic:** - The general equations of motion are evaluated (not integrated) for each position/node contained in the grid *over time*, according to the variation of the positions of both the Sun and the planets. It is important to point out that it is precisely the S/C that would hypothetically occupy each point of the grid (as this is the rationale that motivated the mapping). Nevertheless, it would involve making multiple "portraits".
2) **Static:** - The general equations of motion are evaluated for each position/node contained in the grid at a *single moment*. Thus, in this approach there is nothing in motion and the map will show a "photo" corresponding to a specific instant of time, where each planet occupies a position within their orbits.

The technical obstacles of the first version are enormous, especially in terms of results, i.e, *how to represent a vast number of "pictures"?* (The S/C can be at each grid position within the discretised control volume and at different times, according to the positions of planets). Fortunately, the same preliminary analyses cited before provided the solution: - By randomly choosing any particular epoch where planets remain in a given position, what is visualised is the appearance of PZs[85] exhibiting circular geometric shapes centred on each respective planet. In most cases, with the threshold imposed they remain isolated (without intersection) exhibiting *constant radii*.

4.2.3. Contourplot analyses

Once large matrices containing the result of the evaluation of Equation (2-19) at the nodes of the discretisation have been obtained, the magnitudes of the vector sum of the perturbing accelerations can be isolated and represented together with the dimensions of the control volume (positions of the nodes on X-axis and Y-axis with the origin located at the centre of the Sun and in the ecliptic plane, as a consequence of having adopted the planar projection approach).

Such representation is called contour plot and will include the planetary orbits (with their positions) to differentiate the areas where the perturbing acceleration threshold is being exceeded and where it is not. As mentioned above, regardless of the epoch chosen, the radii of the circular Perturbation Zones centred on each planet (spherical in real space) remain constant in the vast majority of cases, appearing only some exceptions in which two PZs of planets placed in consecutive orbits intersect (not occurring between each pair of consecutive planets but in few cases). It will be discussed in *Chapter 6*.

[85] They would correspond to spheres in real 3D space (remember that planar projection is applied). Regarding the shape, it is important to note that in the end, the perturbations are a result of the distortion of space that each planet causes around it.

4.3. Map validation

Once the tool/procedure is ready, the next natural step is to validate it. To do this, a simulation of a real interplanetary ephemeris trajectory will be performed by numerically integrating the motion equations with the Cowell and Encke perturbation models, presented in *Chapter 3*. The purpose is to demonstrate that no significant difference is when propagating the initial conditions of the ephemeris considering all the planets or simply those selected by the tool. In order to know which bodies are those *selected ones*, it is necessary, firstly, to choose a certain interplanetary trajectory, the preferred one being the one corresponding to the S/C itself that led to the establishment of the perturbation acceleration threshold and, secondly, to apply the tool to obtain the desired information. In other words, 2D maps provide information on *where* the perturbative acceleration threshold is exceeded and which celestial bodies are responsible and should therefore be considered in perturbation models.

Before ending the chapter, **Figure 4-5** shows a schematic flowchart regarding the roadmap that the development and subsequent validation of the tool/procedure requires to meet the objectives set for this thesis. The following three chapters will delve deeper on it part.

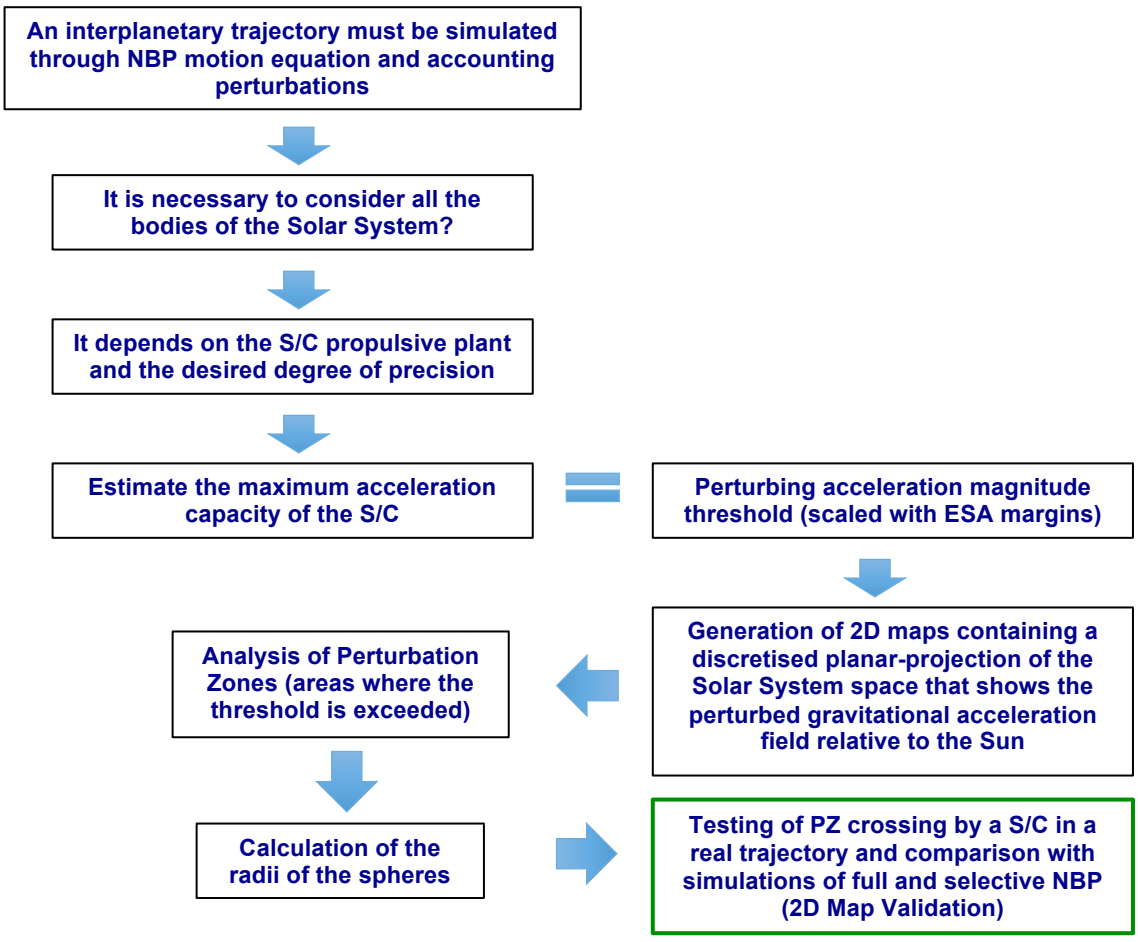

Fig. 4.5 Tool/procedure roadmap outlined in a flow chart.

CHAPTER 5. ESTIMATION OF THE PERTURBING ACCELERATION THRESHOLD

This chapter will promote the first step necessary to apply the tool to be developed. Once introduced the concept of perturbation acceleration threshold – which is the upper limit magnitude of the vector sum of all perturbing accelerations that a propelled object can admit without significantly affecting its motion – a realistic estimate will be made to represent a modern spacecraft. After two iterations, *BepiColombo* is the selected as the reference.

5.1. Maximum acceleration of a spacecraft in the cruise phase

The objective in this section is to estimate the maximum acceleration capacity that a modern spacecraft exhibits during its cruise phase. As already was stated, as representative of modern S/C, an electrical propulsion plant has been chosen. Thus, the strategy will be taking as reference point the *SMART-1 mission* due to it was the first mission that used an electric propulsion system – in particular based on the *Hall effect* – to escape Earth and perform an orbit transfer from a GTO to a final lunar capture [22]. Its scientific importance resides mainly in its preparatory nature for future missions – as *BepiColombo* [23] – that will benefit from primary electric propulsion and deep – space communications [24]. For calculating a first value of the theoretical acceleration that the *EPS* (*Electric Propulsion System*) called ***PPS – 1350 G Hall–effect plasma thruster*** was capable to develop, some technical nominal data will be collected directly from its developer *SNECMA* at a date when that qualified technology was close enough to *SMART-1* launch (it took place at 2003) [25]. **Table 5-1** contains such information and **Figure 5-1** shows a picture of the model.

SMART-1 PPS – 1350G propulsive characteristics			
Discharge voltage (V)	Discharge current (A)	Efficiency (%)	Total Xe flow rate (mg/s)
351	4.28	49.6	5.32

Table 5.1 SMART-1 PPS characteristics**.**

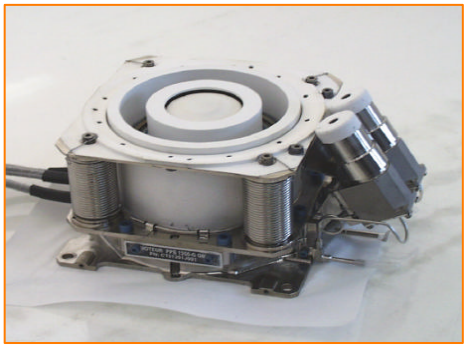

Fig. 5.1 PPS – 1350 G Hall–effect plasma thruster. It can work 7500 h on the ground and 5000 h in flight [26].

The calculation process is simple and uses known theory for space rockets in general and electric propulsion in particular ([27], [B.5]). The applied equations are those followings:

$$\Delta V = I_{sp} g_0 Ln \frac{m_s + m_p + m_e}{m_s + m_e} = V_s LnR \quad \textbf{(5-1)}$$

Where:

I_{sp} is the specific impulse (s)

g_0 is the Earth's gravity at sea level (m/s^2)

m_s is the spacecraft structural mass (payload, tanks and thruster ; kg)

m_p is the propellant mass (Xe ; kg)

m_e is powerplant mass (kg)

On the other hand:

$$V_s = \sqrt{\frac{2\eta P_e}{\dot{m}}} = \sqrt{\frac{2\eta V I}{\dot{m}}} \quad \textbf{(5-2)}$$

Where:

η is the thruster efficiency

P_e is the nominal electrical power supplied to the thruster (W)

V is the nominal electric voltage supplied to the thruster (V)

I is the nominal electric current supplied to the thruster (A)

$\dot{m}$ nominal total Xenon mass flow rate (kg/s)

Substituting the nominal data of PPS-1350G in Equation (5-2) it is possible to compute the velocity to which the propellant (Xe) is accelerated by the electric field once ionised. Next, it is necessary to know the mass budget in order to complete the basic calculations and reaching a value of delta-V. According to [28] the spacecraft dry weight (i.e., m_s + m_e) was 280 kg and the propellant mass was 82 kg (m_p).

Finally, in order to establish an average nominal acceleration that SMART-1 can potentially develop, an additional result will be used. In [26] it can be found a qualification demonstration showing that PPS-1350G is theoretically able to work about 5000 hours in flight. So, Equation (5-3) must be applied. The results obtained are published in **Table 5-2**. It can be observed that is quite large, which can be associated to the large value of the nominal discharge power.

$$a_{nom} = \frac{\Delta V}{t_{propulsion}} \quad \textbf{(5-3)}$$

Nominal SMART-1 EPS capacity	
Delta-V (km/s)	**Acceleration (km/s2)**
4.3	2,39E-07

Table 5.2 Results for estimation.

5.2. Validation of the result obtained

5.2.1. Using data collected post-mission

Table 5-3 collects measured and experimental data achieved post-mission collected by the Propulsion and Aerothermodynamics division of ESA's Directorate of Technical and Quality Management (ESTEC) regarding the SMART-1 electric propulsion system performance [22]. Using this information and the same expressions stated above, it is possible to calculate the real delta-V and subsequently, the acceleration developed. **Table 5-4** shows the results.

SMART-1 Electric Propulsion Performance		
Launch Mass (kg)	Cumulative EPS duration (h)	Average eff. Mass flow rate (mg/s)
370	4958	4.44
Xe mass at launch (kg)	Avg. Discharge power (W)	Average eff. Isp (s)
82	1140	1540

Table 5.3 Measured and experimental data achieved post-mission.

Real SMART-1 EPS capacity	
Delta-V (km/s)	**Acceleration (km/s2)**
3.785	2.12e-7

Table 5.4 Results for validation.

It is possible to observe that the result is significantly smaller than the computed before. It is due to the fact that it is based on collected data after mission, while the previous was just a theoretical guess based on the potential nominal capacity of the EPS. Therefore, *this is closer to the reality and more reliable* (moreover, it is the most conservative). Some conclusions are:

- The order of magnitude for both ΔV and a_{max} matches quite accurately between nominal and experimental data by means of using rocket theory.
- The significant difference is due to the fact that real data was collected post-mission, while the nominal one was just a theoretical guess based on the potential capacity of the EPS based on qualification tests carried out by the developer.
- The main source of discrepancy corresponds to the discharge power. For the nominal it has been considered the qualified status at Beginning of Life (BOL) since the real one is just and average taking into account the physical degradation of the power plant.

5.2.2. Complementary sources. Refinement

Before concluding this section, mention a result included in another source [29], which states that the spacecraft reached ***3.9 km/s***. Additional "double-checks" could be made with results from other papers or data sheets.

5.3. Threshold establishment. ESA's margin philosophy

As indicated in Chapter 4, ESA suggests basing the threshold in terms of a certain fraction of the given propulsion capabilities, rather than directly considering the estimated/calculated value (to be conservative and avoid unexpected problems "on mission").

5.3.1. Iteration 1. SMART-1 based

Therefore, the value to be considered as perturbative acceleration threshold to represent the contours of the magnitude of its vector sum, the reference is, indeed, the most conservative. According to the conclusions achieved in the previous section, it would correspond, in principle, to $2.12 \cdot 10^{-7}$ km/s^2.

However, according to the **R-DV-1**[86] within the published regulations by ESA on its margin philosophy document [30] used for scientific assessment studies, and being even more restrictive (conservative) about *1 % of the maximum capacity* exhibited by the propulsion plant of the S/C will be actually considered as a threshold. Thus, the threshold is formally fixed on ***2.12 x 10^{-9} km/s^2*** for the ***first iteration***.

Note: - In case of considering R-DV-11, the most restrictive ΔV – among all the previous results presented – would be **189.25 m/s**. Considering the same nominal time of propulsion capacity that PPS-1350G can operate in flight (5000 h), it would lead to **1.05139 x 10^{-8} km/s^2**, which is clearly less restrictive.

5.3.2. Iteration 2. BepiColombo based

After establishing the threshold for the perturbative acceleration in the first iteration through the propulsion capabilities of SMART-1, having taken into account the ESA margin philosophy and even added a conservative criterion, it has been considered that a second iteration could be carried out to ensure and/or improve the guarantee of that threshold value.

[86] **R-DV-1** is a Delta-V margin regulation that covers uncertainties in mission design and system performance and will always apply to the Effective Delta-V manoeuvres. Within the document, it can be found a more specific one, the **R-DV-11**, which clearly suggests taking 5 % for accurately manoeuvres and for detailed orbit maintenance manoeuvres.

A large number of papers were reviewed in order to gather information on the performances of the most famous past interplanetary low-thrust missions. **Table 5-5** summarises such information that will be shortly discussed:

Low - thrust missions using EPS							
Mission name	Mass (kg)	Xe Mass (kg)	Total thrust (mN)	Specific Impulse (s)	ΔV (km/s)	Powered flight duration (h)	Acceleration (km/s^2)
SMART-1	**370**	**82**	**67**	**1540**	**3.8**	**4958**	**2.12e-07**
Deep Space - 1	486[17]	72[19] (5)[19]	92[18]	3100[18]	(0.31 computed)	16246[19] (910.3)[19]	(9.58e-08)
Hayabusa[11]	510	66,2	24	3000	2.2	25590	2.39e-08
Hayabusa-2[11]	609	66,5	30	3000	2	13140	4.3e-08
Dawn	1218[14]	425[14] (71.7)[14]	92[15]	3100[15]	(1.8)[14]	51385[16] (6500)[14]	(7.05e-08)
BepiColombo	4074[12]*1	580[12]	290[13]	4300[13]	6.5 [computed]	26000[13]*2	6.94e-08

Table 5.5[87] Low-thrust missions using EPS.

Keys (for *BepiColombo*):

*1 The *Mercury Transfer Module* (MTM, 1843 kg) – equipped with solar electric propulsion system – had to carry two science orbiters: *Mercury Planet Orbiter* (MTO, 1838 kg) and *Mercury Magnetospheric Orbiter* (MMO, 275 + 125 kg)

*2 The mission is currently in progress and its arrival is expected on December 5th of 2025. Thus, it corresponds to a theoretical estimation accounting the technical characteristics of the *QinetiQ T6Kaufman-type* grided ion thruster **[31]**

It is worth clarifying that delta-V's are directly introduced from the source (if found) and are based on characteristics of each propulsion plant. Apart from this, the data between parentheses was published as "*checkpoints*" during the interplanetary cruise. In such cases, the acceleration is preferred to be computed using this information as to be more accurate through accounting just the primary mission (for instance, in DS-1 it corresponds to the 2nd thrusting leg trajectory towards its first asteroid interception).

As for the observations that can be approached with greater relevance:

1) The general trend is a fall in the acceleration as both thrust and total launch mass increase ➔ It might suggest a non-synchronicity between enhanced thrust capabilities and mass increment.

2) Hayabusa missions are a case apart since they exhibit much lower thrusts than SMART-1. It is due to the non-need of achieving a high ΔV since the interplanetary cruise was pretty short (just to intercept an asteroid when its orbit was close to Earth).

3) All missions – except SMART-1 – used higher specific impulses (the double or even almost three times in the case of BepiColombo).

[87] The key for the references is: **DS1** ([32], [33] and [34]). **Hayabusa** and **Hayabusa 2** [35]. **Dawn** ([36], [37] and [38]). **BepiColombo**: [31] and [39].

The first question arising is: *it is possible to understand how this trend works*? To answer this question, a theoretical study will be performed to deepen the relationship between *thrust*, *mass* and *acceleration* in order to explain why the current state-of-the-art shows such a tendency when in carrying out large and long interplanetary missions. Nevertheless, *Hayabusa* missions will not be considered since its concept was in the limit of an interplanetary cruise and would not follow the trend. In fact, they have not experienced an increase in their thrust but a fall. Therefore, hereafter, four spacecraft will participate in this simple qualitative study.

Taking into account these well-known expressions:

$$t_{prop} = \frac{m_p}{\dot{m}} \tag{5-4}$$

$$\dot{m} = \frac{F}{I_{sp}[m/s]} = \frac{F}{g_0 \cdot I_{sp}[m/s]} \tag{5-5}$$

Equation (5-5) relates the *thrust* with and *mass flow rate* with the *specific impulse*. Combining now with Equations (5-1) and (5-3) is obtained:

$$a = F \cdot \frac{\ln \frac{m_s + m_p + m_e}{m_s + m_e}}{m_p} = F \cdot \frac{\ln \frac{m_0}{m_0 - m_p}}{m_p} \tag{5-6}$$

F depends on the state-of-the-art (**thrust technology**)

The second factor depends on the mass performance (**Mass Factor Ratio**)

By examining Equation (5-6) it is noticed that the acceleration can only be increased by means of performing improvements on the thrust. The mass factor ratio always decrease independently of whether the increase happens only at m_0 (by means of increasing m_s, m_e or both) or only at m_p (in such case obviously it increases m_0 in the same extension).

It can be easily checked by calculating each respective limit:

$$\lim_{m_{se} \to \infty} \frac{\ln \frac{m_{se} + m_p}{m_{se}}}{m_p} = 0 \quad ; \quad \lim_{m_p \to \infty} \frac{\ln \frac{m_{se} + m_p}{m_{se}}}{m_p} = 0 \tag{5-7}$$

For the first limit (left) m_p is kept constant and m_{se} increases due to m_s, m_e or both, which any case, is undesired. For the second limit (right) m_{se} is kept constant and m_p increases, which is desired. The big difference hidden in these “zeros” lies in the way the limits decrease.

An analysis is to be conducted were can be observed the different behaviour exhibited by each of them when Δm_{se} = 1 kg and when Δm_p = 1 kg[88] (i.e, when MFR is decreasing as a consequence of increasing one or the other with increments of 1 kg). Some preliminary considerations must be taken into account:

1) Departing from actual SMART-1 values (m_{se} = 288 kg and m_p = 82 kg)

2) Computing values of mass factor ratio for +1kg increments in each case (keeping the original value for the other)

3) Normalization to zero for comparison of the effect of each increment (so we can simply refer to the cited mass increment as Δm)

Figure 5-2 shows the results[89]. The blue curve exhibits much less damping, being considerably more exponential, which leading to a maximum for certain Δm (in SMART-1 case, it is placed at Δm = 690 kg). This can be better observed in **Figure 5-3** that shows the difference between both curves (simply obtained by subtracting the smallest to the largest one).

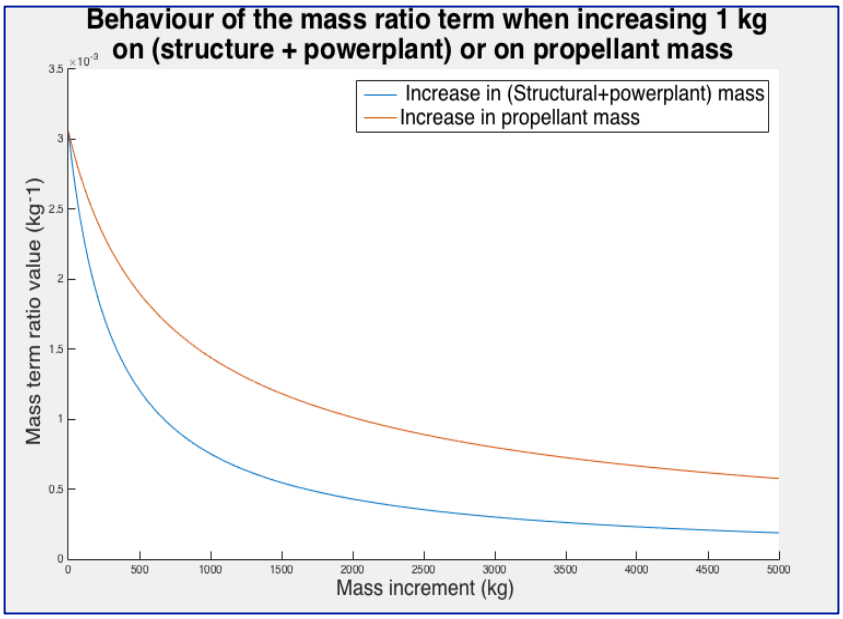

Fig. 5.2 Behaviour of MFR term when increasing 1kg on (structure + powerplant) or on propellant mass.

In a real situation, it is not possible keeping constant m_{se} while increasing m_p due to the need of increasing the thickness of propellant tanks (or even changing its material), which affects to m_s (in principle it is possible to assume that powerplant mass remains unaltered). Therefore, since the current state of the art will imply $\Delta m_s = x\Delta m_p$ (thus not remaining constant) *the actual curve must be placed somewhere in between.*

[88] It refers to increments from kg to kg (it is in the end simply incrementing)

[89] In **Appendix B.3** is placed the script used to obtain the plots (except the fits, made I excel)

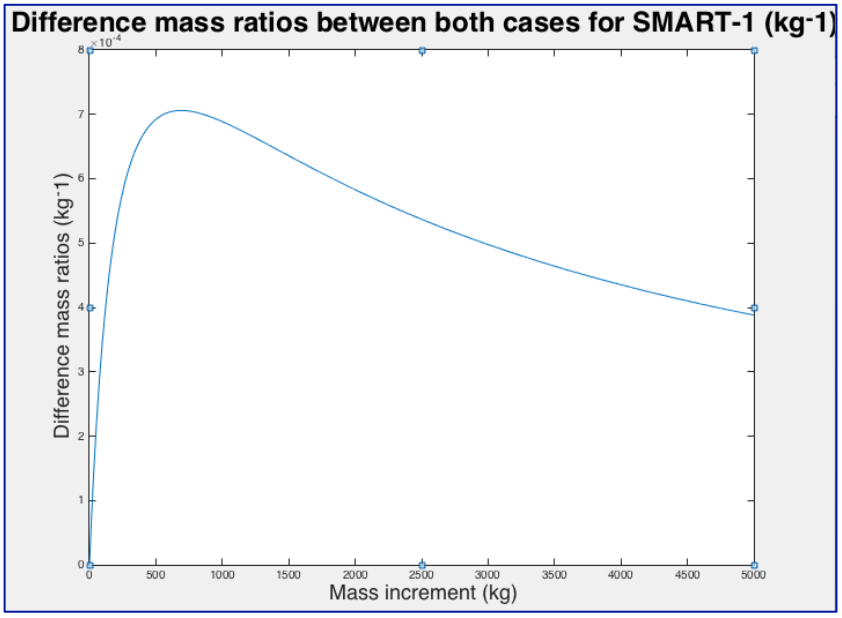

Fig. 5.3 Difference between MFRs (variation in the behaviour between Δm_{se} or Δm_p).

On the other hand, another question is: *what happens for the rest of missions?* By applying the same analysis but departing from the m_{se} and m_p values exhibited by each of these missions, what is achieved is the difference[90] in behaviours provided by **Figure 5-4.** In addition, **Table 5-6** shows the values of maximums and the Δm corresponding each.

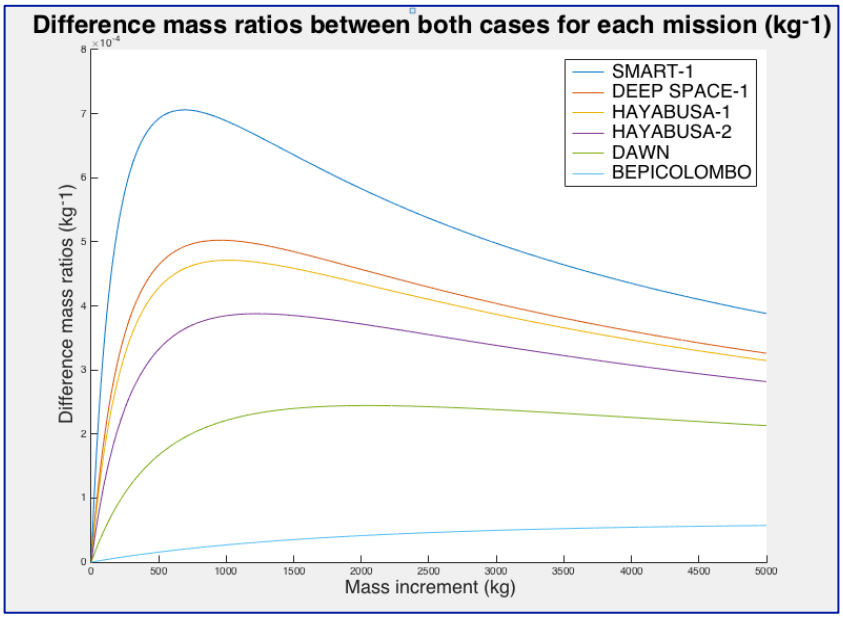

Fig. 5.4 Difference between MFRs for all missions (variation in the behaviour between Δm_{se} or Δm_p).

[90] The graph presented by **Figure 5-4** is analogous to the one at **Figure 5-3**, but computed not only for SMART-1 but for the rest of missions.

	Maximum (kg^{-1})	Increment (kg)
SMART1	7,060E-04	690
DS1	5,025E-04	956
HAY-1	4,713E-04	1015
HAY-2	3,878E-04	1229
DAWN	2,444E-04	2054
BC	5,965E-05	8042

Table 5.6 Maximums in the differences and Δm.

Another ancillary representation can be useful to confirm the further analysis. **Figure 5-5** shows how the maximums decrease with Δm.

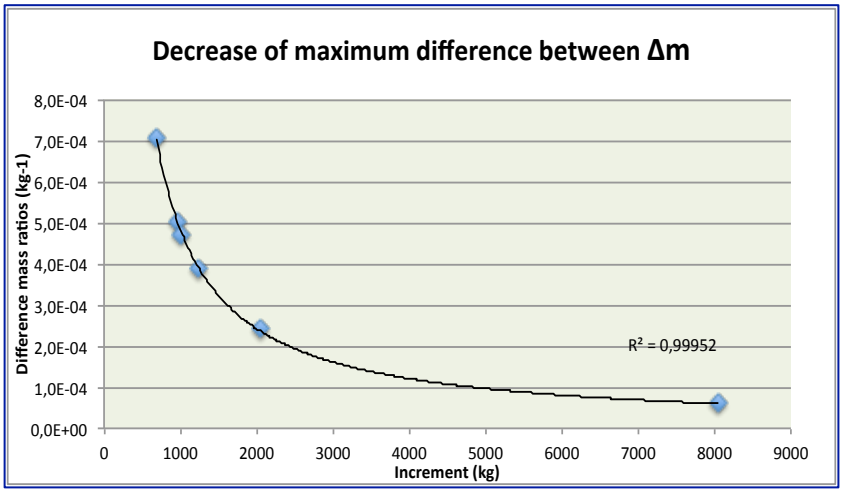

Fig. 5.5 Decrease of maximum difference between mass increment.

It indicates that, as m_{se} becomes larger[91] the maximum decreases according to a square-powered function (as it shows the R^2). Moreover, the maximum suffers a displacement towards higher Δm positions. It indicates that, *the* smaller m_{se} the greater penalisation if the total mass at launch of the spacecraft increases due to the structure/powerplant rather than just adding propellant (the bigger masses take benefit of a "*damping effect*", so penalises less). These results can be summarised in some points:

- Any increase in spacecraft mass – independently of its nature – penalises its acceleration if the thrust does not increase in a certain way *(inversely to mass ratio)*
- As expected, to increase only structure/powerplant mass reduces faster the value of the mass factor ratio than to increase m_p, since it exhibits a less damped behaviour
- Small spacecraft (low m_0) are more sensitive to such penalty as evidenced by facts:
  - The maximum difference in the mass factor ratio decreases as m_0 increases.
  - The decrease follows a negative squared-power trend, which indicates that the larger the spacecraft, the larger the mass increment needed to reach such maximum

[91] The m_{se} of each mission (before performing any mass increment). It is referred as to allow distinguishing behaviour between each serie (i.e, mission) since depending on the initial m_{se} and m_p their behaviour when Δm varies.

- An immediate consequence is that, for a given Δm, small spacecraft need a higher ΔF than the larger ones to maintain the original acceleration
- It is still pending the analysis of how has been the thrust performance in missions

Awaiting for the analysis suggested by the last point, a preliminary conclusion would be the next: in case that an increase on m_{se} must be done for an already designed spacecraft (e.g, payload based on scientific purposes, life support systems…etc) it will suffer a drop in the acceleration capacity more damped as larger be the m_{se}.

Analysis of the state of the art based on the missions cited

Until now, all previous analyses were performed based on theory applied to data regarding the characteristics of each mission (essentially masses)[92]. However, let us to analyse now the truly behaviour performed by those actual missions. To do so, three considerations must be taken into account:

1st) *Normalising* with the SMART-1 reference values:

$$\frac{a_{mission}}{a_{SMART-1}} = \frac{F_{mission}}{F_{SMART-1}} \frac{MFR_{mission}}{MFR_{SMART-1}} \quad \textbf{(5-8)}$$

In order to maintain the original acceleration, the product of dimensionless ratios should be 1, which means that $F_{mission}$ should be increased in such a way that allows compensating the sure decrease that $MFR_{mission}$ suffers with respect $MFR_{SMART-1}$

2nd) As mentioned before, Hayabusa missions will not be considered since they have not experienced such increase in their thrust but a fall.

3rd) A simple representation consisting in plotting $F_{mission}/F_{SMART-1}$ vs $MFR_{SMART-1}/MFR_{mission}$ will be carried out as to analyse the deviation exhibited with respect a linear trend (and compared to what it should be if were a function of type y = x)

The previous acceleration values for each mission were computed dividing its ΔV – taken directly from the literature or estimated through applying the Tsiolkovsky's rocket equation and their propulsive characteristics – by the total powered flight duration. Nevertheless, if compared those values with the ones obtained by means of using the recently developed theoretical expression, a little variation occurs, as **Table 5-7** shows.

Low - thrust missions using EPS							
Mission name	Mass (kg)	Xe Mass (kg)	Total thrust (mN)	ΔV (km/s)	Powered flight duration (h)	a = ΔV/t_p (km/s^2)	a = F.MFR (km/s^2)
SMART-1	**370**	**82**	**67**	**3.8**	**4958**	**2.12e-07**	**2,04712E-07**
Deep Space - 1	486[17]	72[19] (5)[19]	92[18]	(0.31 computed)	16246[19] (910.3)[19]	(9.58e-08)	2,0733E-07
Dawn	1218[14]	425[14] (71.7)[14]	92[15]	(1.8)[14]	51385[16] (6500)[14]	(7.05e-08)	9,28967E-08
BepiColombo	4074[12]*1	580[12]	290[13]	6.5 [computed]	26000[13]*2	6.94e-08	7,67891E-08

Table 5.7 Comparison between accelerations developed theoretically (right) and using literature (dividing known delta-V by known time).

[92] i.e., still remain the question about how the thrust (F) behaved in the trend compared to the mass.

In the case of SMART-1 it almost coincides due to the fact that the mission consisted barely on reaching the moon and crashing the spacecraft. Thus, its flight duration was accurately computed. In DS1 and DAWN, that time was described as to achieve the ΔV for a specific cruise section. In addition, they were missions planned to get more than one destination, leading to a possible extra hours accounted. In the BepiColombo mission, the powered flight duration is barely a guess, as it is logical.

Continuing with the analysis pursued, if theoretical accelerations are taken as reference (since they reflect their real capacity based on propulsive characteristics) and carry out the 3rd step introduced earlier a fit of the data is represented in **Figure 5-6**. It is observed that:

- In despite of the criticality of mass increase in S/C with low m_0 (as seen previously), DAWN copes with that due to a sufficient increase on its thrust with respect SMART-1 ➔ ***ΔF/Δm₀ ≈ 216 nN/kg***
- $a_{S/C}$ decreases in the rest of missions due to a insufficient increase on thrust to compensate the non-linear fall on MFR ➔ ***ΔF/Δm₀ << 216 nN/kg***
- Nevertheless, data seems to fit a polynomial regression suggesting a future intersection with the theoretical fit leading to recover $a_{SMART-1}$, which is agree to the fact of being such reduction more and more slowly

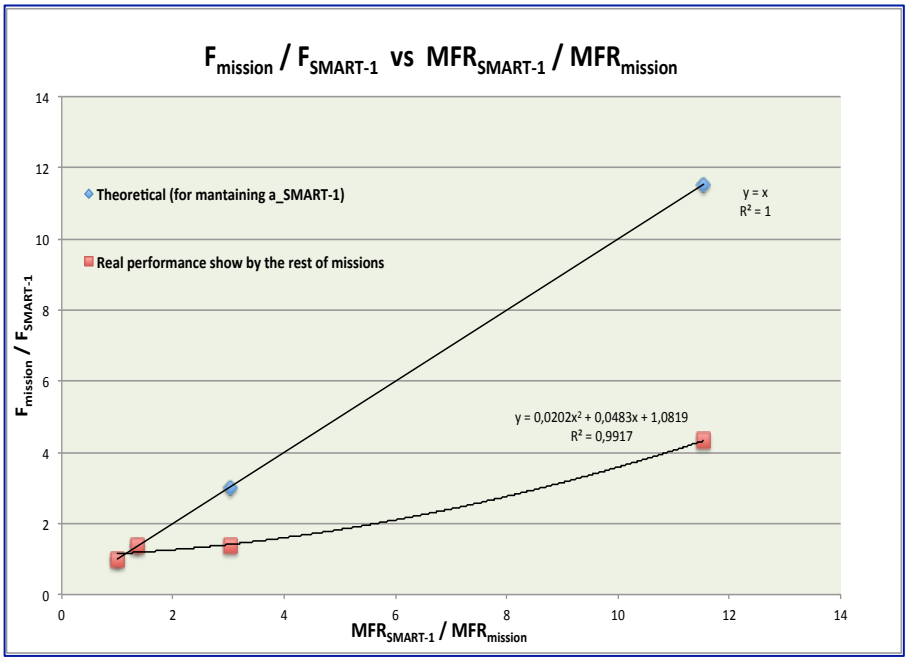

Fig. 5.6 Mathematical fit of ratios to evaluate the trend followed by mass and thrust.

Figure 5-7 shows both curves extrapolated. An intersection occurs at 46.1085 (equivalent to a slightly more than $m_0 \approx 17000\ kg$).

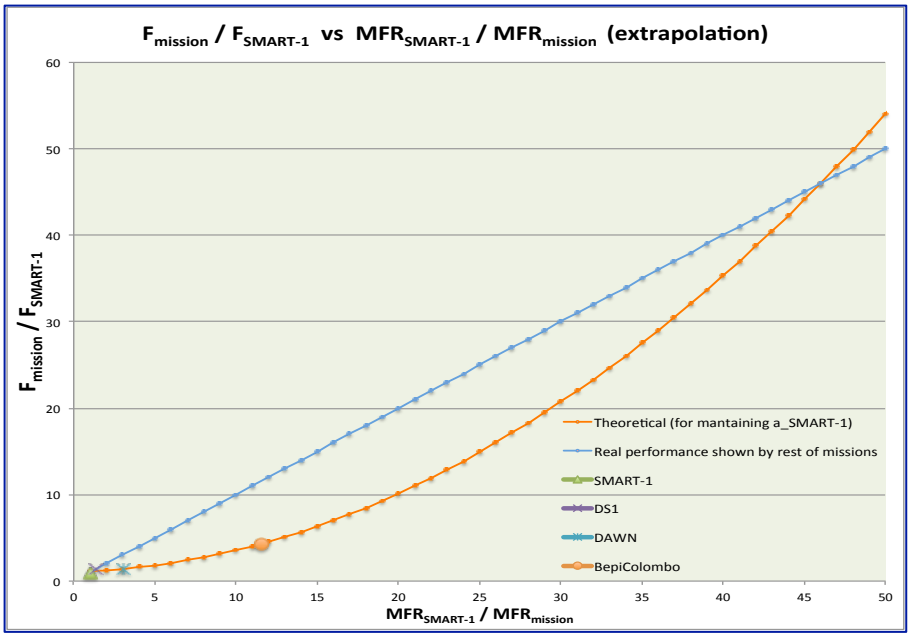

Fig. 5.7 Mathematical fit + extrapolation.

The polynomial fit suggests that due to the damped fall of MFR – seen previously– the current $\Delta F/\Delta m_0$ trend (state of the art) is sufficient to stop the $a_{mission}$ decrease and recover after or even overcome $a_{SMART-1}$ at some point when m_0 is high enough. Summarising the overall conclusions achieved by all the analysis carried out within this section:

- Taking as a reference the smallest spacecraft developed for an interplanetary mission using electrical propulsion, it has been seen that, in principle, the current state of the art could not maintain acceleration while increasing the launch masses
- The understanding has required a quite deep analysis, which consisted of developing a theoretical expression that relates F with m_0 and m_p
- Regardless of the nature of mass increase – being able to be just the inert mass $m_{se,}$ the propellant m_p or both (the most realistic) – the Mass Factor Ratio always decreases. In addition, it does it in a more damped way only increasing m_p (desirable)
- A relative function ($MFR_{\Delta mp} – MFR_{\Delta mse}$) was computed to see how it behaved in a worst-case scenario (that is, what would happen if instead of increasing m_p a pure increase on m_{se} were carried out). It showed the existence of a maximum whatever the mission was
- These maximums evidenced that small spacecraft are highly penalised if they increase m_{se} rather than m_p. It was proven by observing that they tend to decrease with the launch mass and by the fact that it fitted to a squared – power function, damped as Δm increases
- The dimensionless fit agreed those previous analyses and showed that acceleration is decreasing slower every time, forecasting a recover of SMART-1 level (even

overcoming it) for very high launch masses if the current average $\Delta F/\Delta m_0$ *is kept (state of the art)*

To end this chapter, the last remaining thing is to establish a definitive threshold and to scale it according to the ESA's margin philosophy document and the additional conservative criterion. Taking everything into consideration, the perturbative acceleration threshold is **6.94×10^{-8} km/s^2**, which applying after the ESA margin philosophy document **R-DV-1** corresponds to **6.94×10^{-10} km/s^2**.

The justifications are based on the results and conclusions from the thrust – total launch mass analysis:

1) The sample of data is not really high, which does not lead to a very robust statistic result. However, is based on the published interplanetary missions using electrical propulsion.

2) Hayabusa missions were neglected since they did not follow the trend of increasing the thrust while increasing the launch mass (they were not real interplanetary trips).

3) Mass Factor Ratio decreases uniformly both with increase on structure/power plant and propellant, being more damped the fall corresponding to the last one.

4) The relative MFR function showed the existence of maximums that decrease with the launch mass in a damped squared – power trend, which means that small spacecraft are much more penalised when the increase happens on m_{se} instead of m_p.

5) The current state of the art can cope with the acceleration reduction, which happens due to the non – linear MFR fast decrease for small spacecraft, but which will be sufficient for high launch masses in the future if it is kept.

All these points clearly justifies the use of a threshold based on the BepiColombo mission, since it could be the one with the minimum value of acceleration on the modern age *(Hayabusa are not realistic and hypothetical manned missions with much more launch mass will be significantly less penalised in terms of MFR, so the current improvement trend on F/m_0 should lead to higher accelerations)*

CHAPTER 6. MAPS OF PERTURBED SUN GRAVITATIONAL FIELD

Throughout this chapter, the mapping of control volume will be implemented in software to obtain the two-dimensional maps. These maps, which represent the region of the Solar System under study, will be analysed by examining the areas in which a spacecraft, accelerated with respect to the Sun, would be subject to planetary perturbations above its acceleration threshold. This will make it possible to establish a criterion for the identification of the relevant disturbances in a given trajectory.

6.1. Software implementation

The aim for obtaining two-dimensional maps of the region of the Solar System under study lies in the fact that its vector field, although complex, encloses vast information about nature of motion. The calculation of the instantaneous perturbing acceleration within Sun's gravitational field will allow predictions to be made about which celestial bodies actually interfere with the motion of an artificial object, and must therefore be compulsorily considered[93]. This a more convenient fashion of understanding the nature of deviations when propagating a trajectory, rather than examining distances, since acceleration needs some application time to cause velocity variations and, ultimately deviations.

Figure 6-1 schematically shows the functioning of the main algorithm[94] each time that is executed. It *creates a control volume* (discretising the region under study in a grid of positions) *evaluates in each node the general equations of motion* given by Equation (2-19) developed in *Chapter 2* and, finally, *performs the contourplots* that allow interpretations to be made. Two comments must be made:

- Real orbits in space have three components in both position and velocity but according to the foundations of this tool, the control volume should be a 2D region. Thus, the planar projection is applied by rotating an angle equivalent to that formed by the angular momentum of each orbit with the Z-axis through the axis that points towards the node line. Then, the Z components are very small and can be approximated to zero (it moves in tens to hundreds of kilometres). **Figure 6-2** shows this action with one of the orbits.

[93] Implicitly, it leads to the knowledge of those that can be neglected without loss of accuracy.

[94] Two maps will be obtained, called ***Window 1*** (containing Mercury, Venus, Earth and Mars) and ***Window 2*** (containing Window 1 plus Jupiter and Saturn). Instead of evaluating the equations of motion within the orbit of a tested planet (the code is also prepared for that) by centring on the outermost orbit (Mars in Window 1 and Saturn in Window 2) the inner ones are automatically included without the need to create excessively large grids. In this way, more nodes can be introduced to obtain a finer grid (more precision).

- To create the domain or window of positions, the "grid" (X and Y) is constructed that goes from the minimum values of the flat orbit to the maximums[95] (slightly enlarged above/below and right/left.). The step is selectable (the smaller the step, the finer).

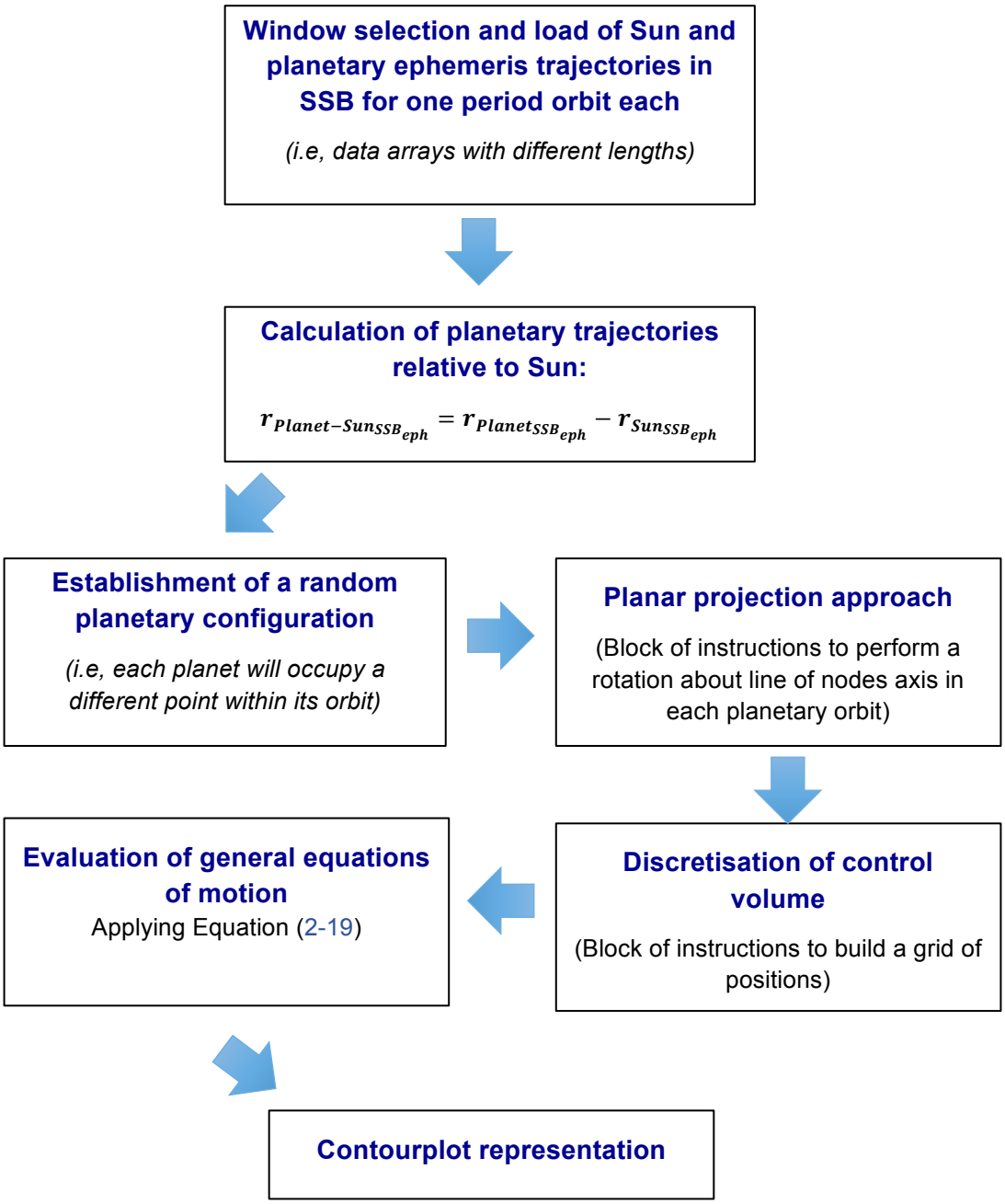

Fig. 6.1 Flowchart schematising the execution of the main algorithm.

[95] This corresponds to the extremes delimited by the major and minor axes, respectively, of each orbit.

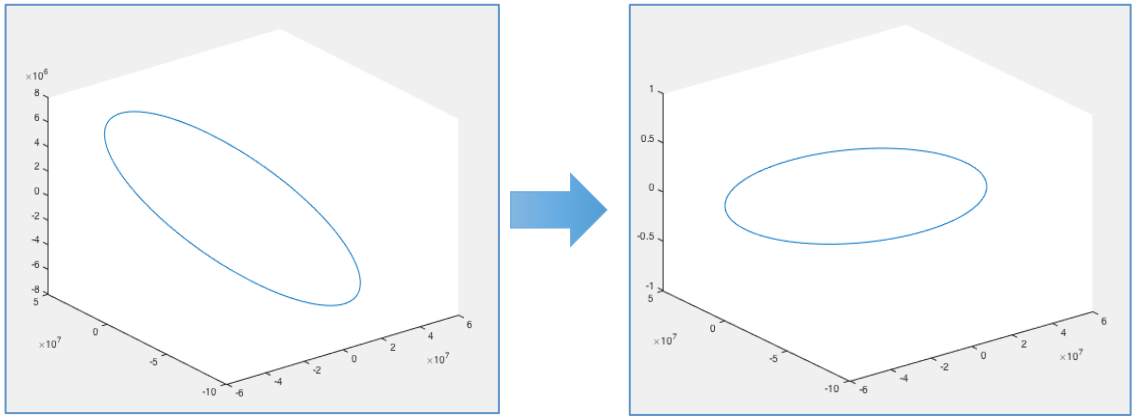

Fig. 6.2 Rotation about the line of nodes an angle equal to the angle formed by the angular momentum of the orbit with the Z-axis of the inertial frame.

Note: - The main algorithm and ancillary functions are implemented in **Appendix B.4**.

6.2. Discussion of results

Before jumping to the results obtained, remember that what is expected is essentially that the 2D map shows the circumferences around each planet where the disturbance threshold is exceeded. As discussed in *Chapter 4*, perturbations come from the fact that each planet gravitationally distorts the space around it, so it could happen that either these perturbative spheres are *isolated*, or that they *intersect* or even that the perturbations of the massive outer planets "*engulf*"[96] the orbits of the interiors

6.2.1. Preliminary considerations

Some important preliminary considerations need to be stated:

1) *Window 1* is centred in Mars and *Window 2* is centred in Saturn (see **Figure 6-3**)

2) It is chosen in this way to be able to appreciate more detail in the maps, bearing in mind that the special interest is in the Earth as point of departure.

3) Window 1 has dimensions of X,Y in the range $[-2.5,2.5]\cdot10^8$ km^2 or [-1.67,1.67] AU. Window 2 $[-1.5,1.5]\cdot10^9$ km^2 or [-10.027,10.027] AU (step of $1\cdot10^5$ km)

4) Since the orbits are not concentric to each other (especially in some cases) we will also examine what happens when the position vectors of two consecutive planets are aligned in the direction of the minimum orbital distance.

[96] Fortunately, it did not happened for the set threshold. Otherwise, it would have compromised the power of the tool/procedure. E.g, imagine that for the threshold set for BepiColombo, the PZ of Jupiter would have “engulfed” all internal orbits. This would directly mean that Jupiter should be always be considered, regardless of the trajectory being designed (and would made it difficult to differentiate where the other PZs associated with the inner planets would be located).

5) The planets will be represented magnified but in approximate proportion between them, so that they can be appreciated in the 2D maps (since with the threshold obtained, the zones of perturbation produced by each planet are "too big[97]").

6) The threshold to be used is the calculated as 1 % of the maximum acceleration that BepiColombo mission can give: B ≥ 0.01; $a_{threshold} = 6.94 \cdot 10^{-10}$ km/s^2, according to ESA's margin philosophy guideline *R-DV-1* (see *Chapter 4*).

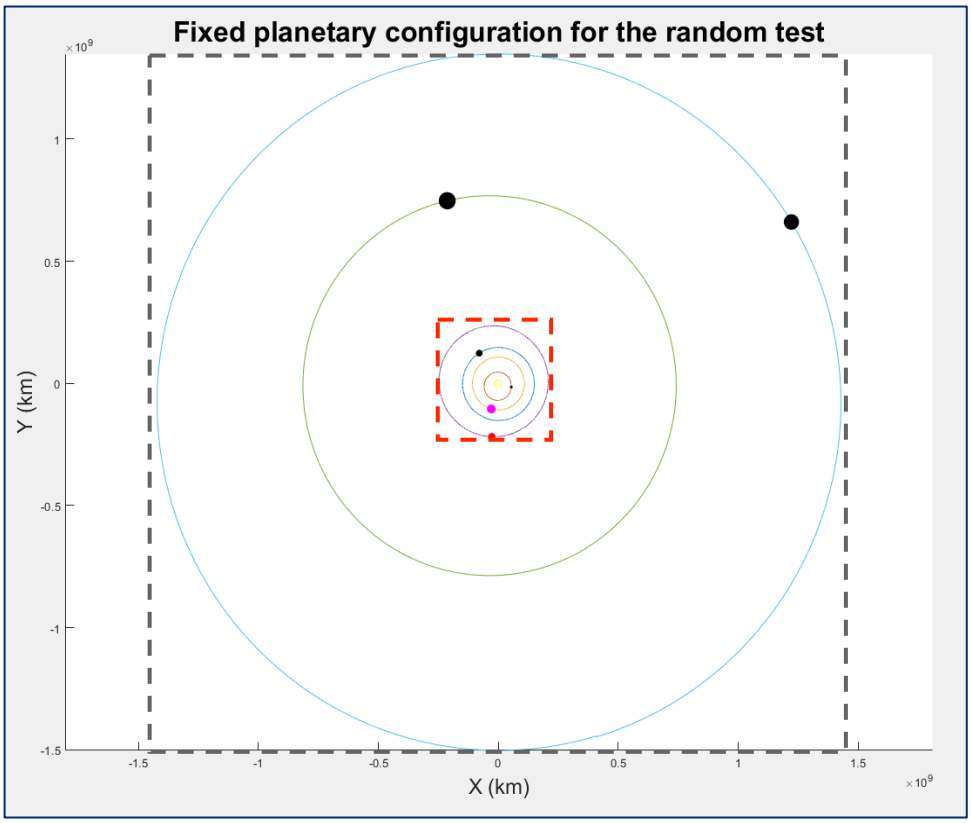

Fig. 6.3 Volume of Control representing the Solar System under study. It is divided into: Window 1 (black grid): Mercury, Venus, Earth and Mars. Window 2 (red grid): Previous planets plus Jupiter and Saturn.

6.2.2. 2D Maps

Since *two extreme cases* have been detected, they will be introduced following the order of performed analyses.

Window 1 (Mercury, Venus, Earth and Mars)

The first case detected corresponds to the random planetary scenario where the planets are not aligned, being the most likely.

[97] This may be a comment subject to interpretations. There exist limitations on this tool/procedure developed, as will be explained at the end of the chapter. Some of them may even be the selected threshold (in case it is considered high). However, here the term "too big" only referred to the magnification of the planets in the maps where their PZs are much bigger.

As it can be observed at **Figure 6-4** the yellow circles (spheres in real space) correspond to the areas of the orbital plane where the threshold is exceeded. These critical areas are clearly isolated. Thus, *in no area of the map is the threshold exceeded by the perturbing action of more than one planet*.

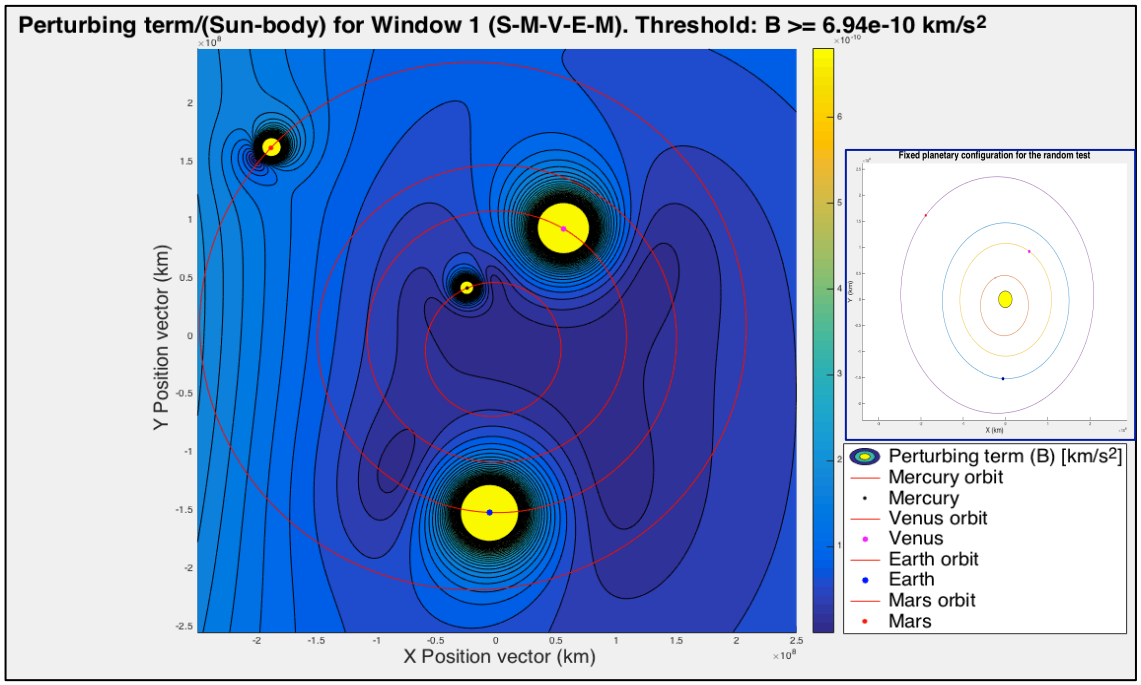

Fig. 6.4 2D Map for Window 1 + Planetary configuration scheme.

Looking at the image, one can easily wonder what the contribution of the Moon is on that map, since it is within the Earth's Perturbative Zone. As the main application of 2D maps is for asteroid rendezvous with the S/C departing from Earth, the passage through Moon's PZ with the set threshold will almost certainly take place (the rest of the natural satellites have been omitted). Although its gravitational action has been included in the calculations of the algorithm, with its corresponding term in the sum of perturbations in the general equations of motion, it is necessary to remove the Earth to see its area of perturbing influence (see **Figure 6-6**).

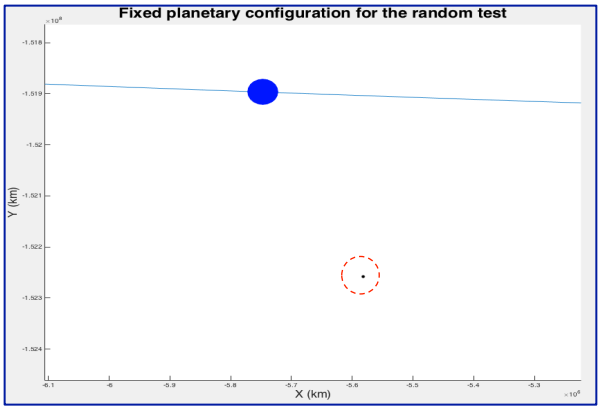

Fig. 6.5 Magnification of Earth's orbit to see the Moon. For the performed execution it is at $3.975 \cdot 10^5$ km of distance from Earth.

Figure 6-6 shows a replica of the 2D Map published in **Figure 6-4** but removing the presence of the Earth. So, what it remains at that location is just the Perturbation Zone created by the Moon.

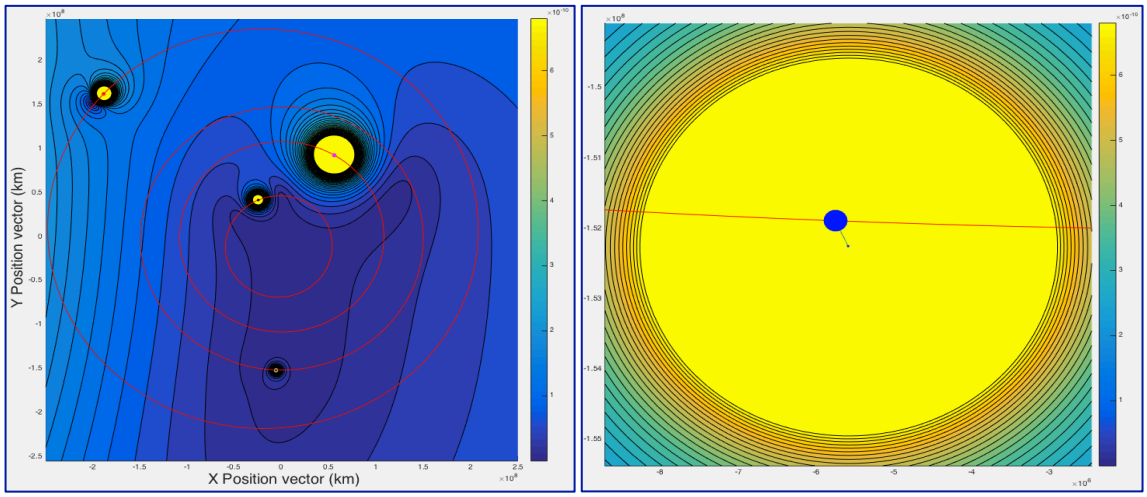

Fig. 6.6 2D Map for Window 1 showing Moon's PZ (amplified on the right). The red line in the picture of the right corresponds to Earth's orbit and the planet in blue is the Earth.

It is possible to observe that Earth's orbit passes through the Moon's disturbance zone not far from the middle. The Moon produces a disturbance zone much smaller than that of the Earth. To make ourselves an idea, at an altitude of 35786 km - in the direction that joins their centres - its value is $3.64 \cdot 10^{-8}$ km/s^2 as opposed to the $3.20 \cdot 10^{-4}$ km/s^2 produced by the Earth by itself (almost four orders less). In any other direction, that disturbance decreases, however the Moon travels around the Earth about 13°/day (quite fast), and an electric propulsion spacecraft such as the BepiColombo mission increases its speed very slowly. As a conclusion of this extra analysis, *for the threshold set, from an altitude GEO - whatever the direction the S/C departs to - the action of the Moon should start to be taken into account*.

Having studied the general case in which no planetary alignment occurs, it is interesting to examine what would happen if an alignment occurs at the minimum distance possible within the orbits (taking into account the size of the PZs, this test it is only necessary for neighbouring planets). **Figure 6-7** shows the case where an alignment between Venus and Mercury occurs (which is not likely, but still possible).

What can be observed is that, taking into account that the orbits of Mercury and Venus are not concentric, even at point of maximum proximity (≈ 0.26 AU) their areas of disturbance do not produce an overlap. It is important also to note that, the black contours - which do rub against each other - are areas of accumulation of mesh elements where there is a great variation in the disturbance, but the threshold is not exceeded. Thus, *there is no any issue coming from a possible interaction between PZs of Mercury and Venus*.

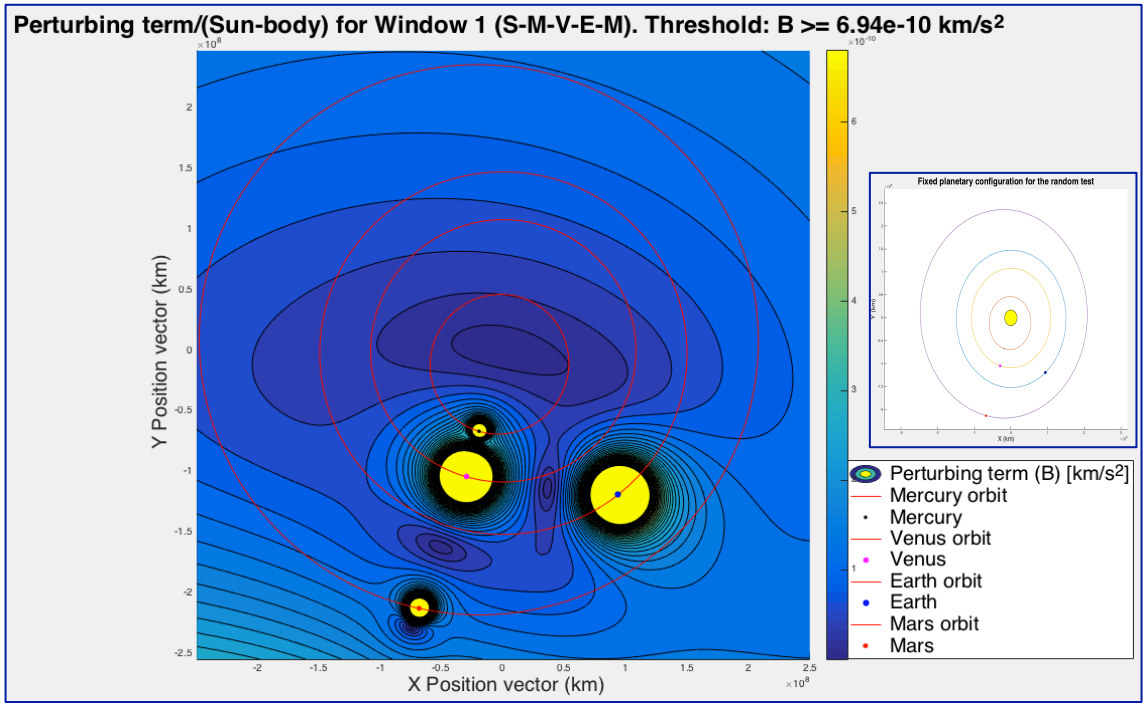

Fig. 6.7 2D Map for Window 1 + Planetary configuration scheme. Case of alignment between Mercury and Venus.

Figure 6-8 shows the results of the case of alignment Venus - Earth at a minimum distance. In this case, it is observed how, apparently, the zones of influence of Earth and Venus *overlap producing a destructive interference*, since the circles do not appear complete. The orbits of Venus and Earth are quite concentric, which means that any alignment (which occurs every 592 days approximately) would produce that interference.

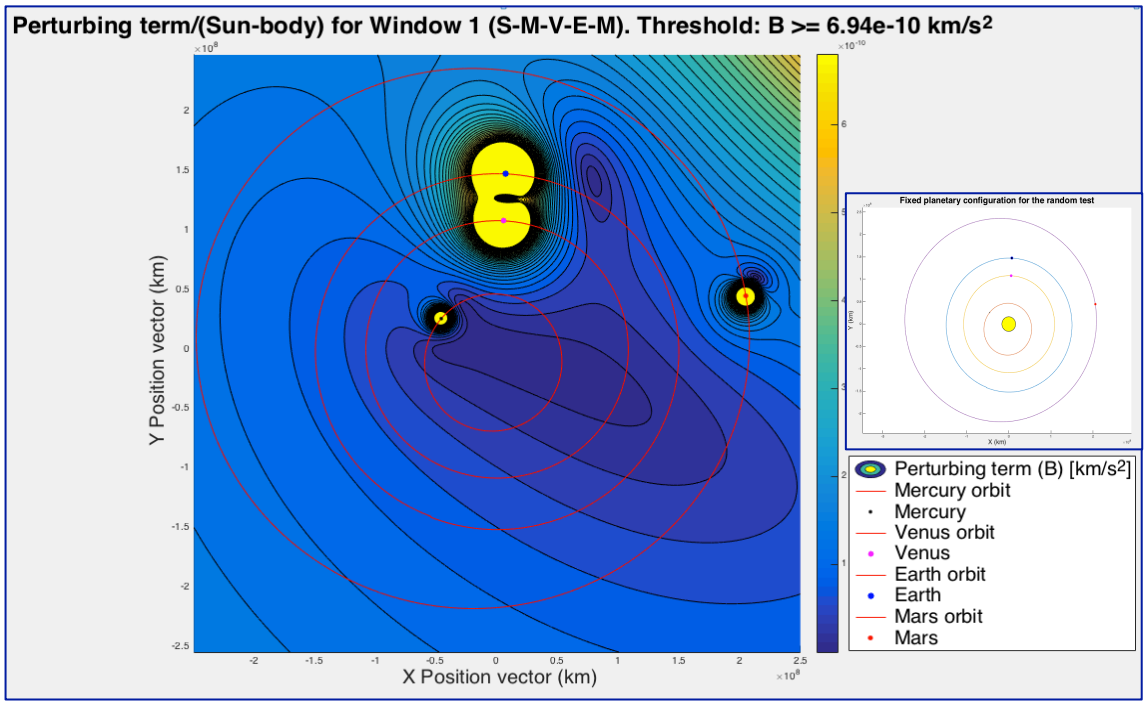

Fig. 6.8 2D Map for Window 1 + Planetary configuration scheme. Case of minimum distance alignment between Venus and Earth (≈0.264 AU).

To better observe what is happening, a magnification of the 2D Map is shown in **Figure 6-9**.

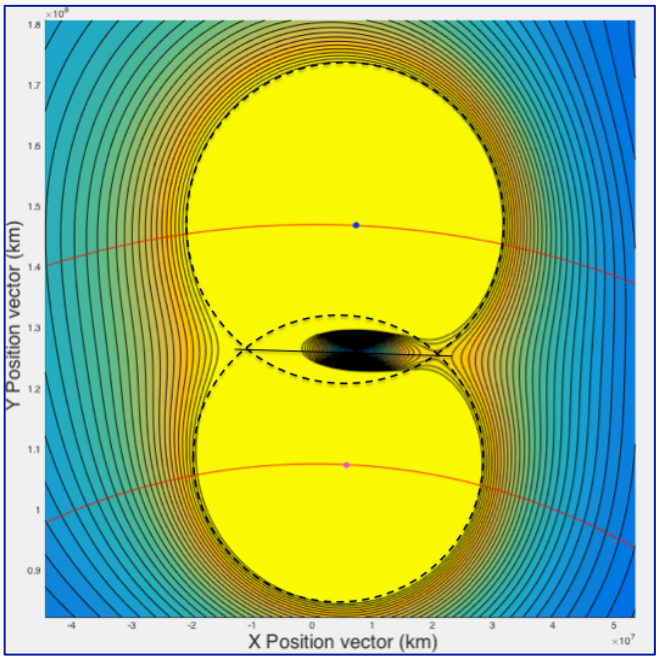

Fig. 6.9 Magnification of the intersection of PZs in the case of minimum distance alignment between Venus and Earth.

Entering into a discussion of the picture, the geometric circumferences in strokes correspond to those that planets would have in scenarios where there is no interference (general case). The ratio between circumference sizes is 1.12:1 for the Earth, which more ore less coincides with being about 1.23 times more massive. The combined disturbance area - produced by the superposition of the individual PZs - exhibits a high symmetry, although not complete since, for reasons of discretisation, the planets are not perfectly aligned. The intersection of the auxiliary circumferences produces two circular sectors, leaving the ellipse associated with the destructive interference zone divided into approximately equal halves (more surface on the Earth's side, according to the proportion).

The preliminary conclusion in this case is basically that it is confirmed that the circumferences of both planets overlap, giving rise to *areas where interaction is constructive and areas where it is destructive*. Nevertheless, a further analysis can still be made. **Figure 6-10** contains a graph where it is represented the magnitude of the perturbation term in front of the length travelled along the line[98] that joins the points of intersection between the auxiliary geometric circumferences (see **Figure 6-9**).

[98] That line travels along the “destructive” section that appeared as a consequence of the interaction/overlap of the Perturbation Zones of both planets.

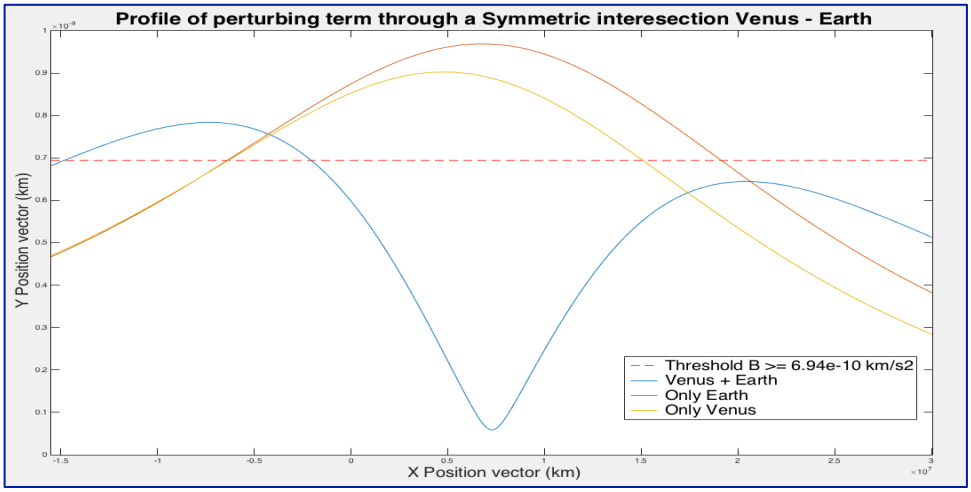

Fig. 6.10 Profile of perturbing term through a symmetric intersection (Venus-Earth).

Due to the approximately symmetrical division, the disks of each planet intersect in the same X position when they are one in absence of the other, therefore the profiles overlap. The subsequent deflection is due to the difference in mass and, therefore, to the different maximum perturbation reached in their PZ. In the combined profile there is a clear synergy, since in none of the individual profiles is the threshold exceeded until well advanced in position. At a certain position, the perturbation magnitude falls from the threshold and no longer recovers at any time (including a strong depression). The main conclusion for this case is that *in trajectory propagations through the destructive interference section, 2BP could be used (advantage), but in the constructive where the threshold is exceeded it would require taking into account four bodies (drawback)* being the combined perturbation level greater than the individual.

To finish the analysis corresponding to Window 1, **Figure 6-11** shows the results for case Earth - Mars on alignment at a minimum distance.

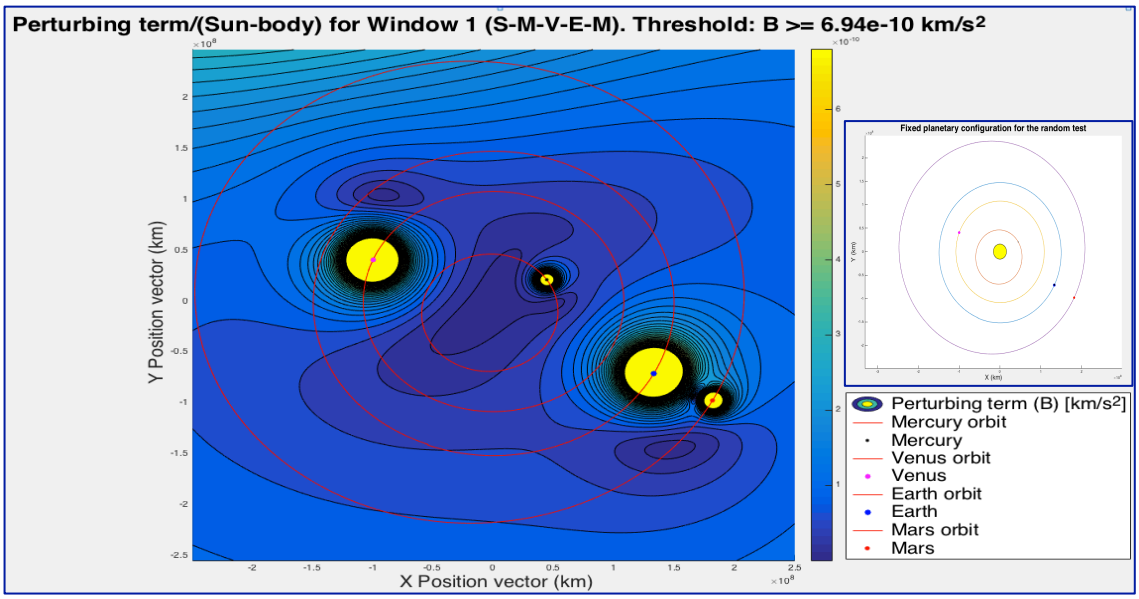

Fig. 6.11 2D Map for Window 1 + Planetary configuration scheme. Case of minimum distance alignment between Earth and Mars (≈0.37 AU).

The orbits of Earth and Mars are highly non-concentric, which means that this scenario does not occur more than every 250755 days ≈ 687 terrestrial years (minimum common multiple between the sidereal periods of Earth and Mars). However, as already happened in the case of Mercury - Venus, *even at this point of maximum proximity their areas of influence do not produce an overlap* (isolated circumferences).

Window 2 (Mercury, Venus, Earth, Mars, Jupiter and Saturn)

The same analyses will now be presented as before. **Figure 6-12** shows the results obtained when running the code one more time to create a new random planetary distribution. What can be seen is that, as in Window 1, the critical zones are isolated - those of Jupiter and Saturn are enormous. *The threshold is always exceeded due to the action of a single planet*. Jupiter's orbit is quite concentric with those contained in the inner window, but not so with Saturn's which orbit is almost invaded. Moreover, Jupiter's PZ is somewhat flattened due to the compression of the iso-lines of its gravitational field, probably due to the configuration obtained from the inner planets.

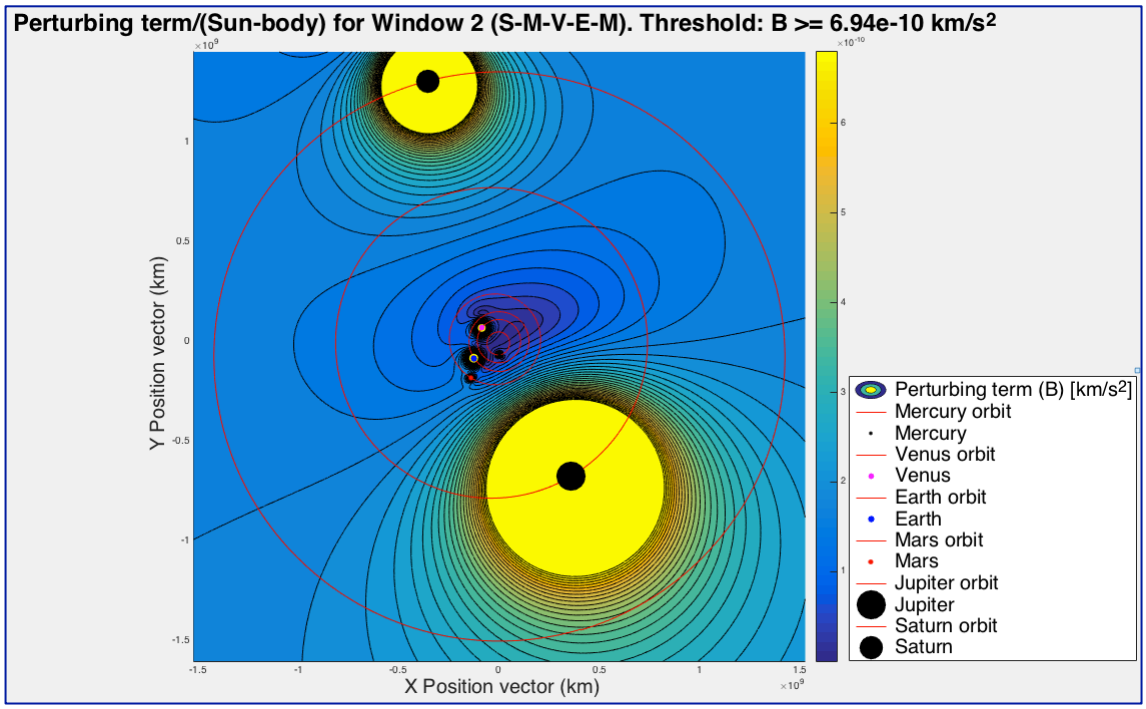

Fig. 6.12 2D Map for Window 2. General case (no alignments).

On the other hand, when an alignment of minimum distance between Jupiter and Saturn is simulated, it is observed that, as happened with the system Venus – Earth, their Perturbation Zones overlap/merge to give one where the threshold is not reached (nonetheless, the geometric addition should cover everything). It is reflected by **Figure 6-13**.

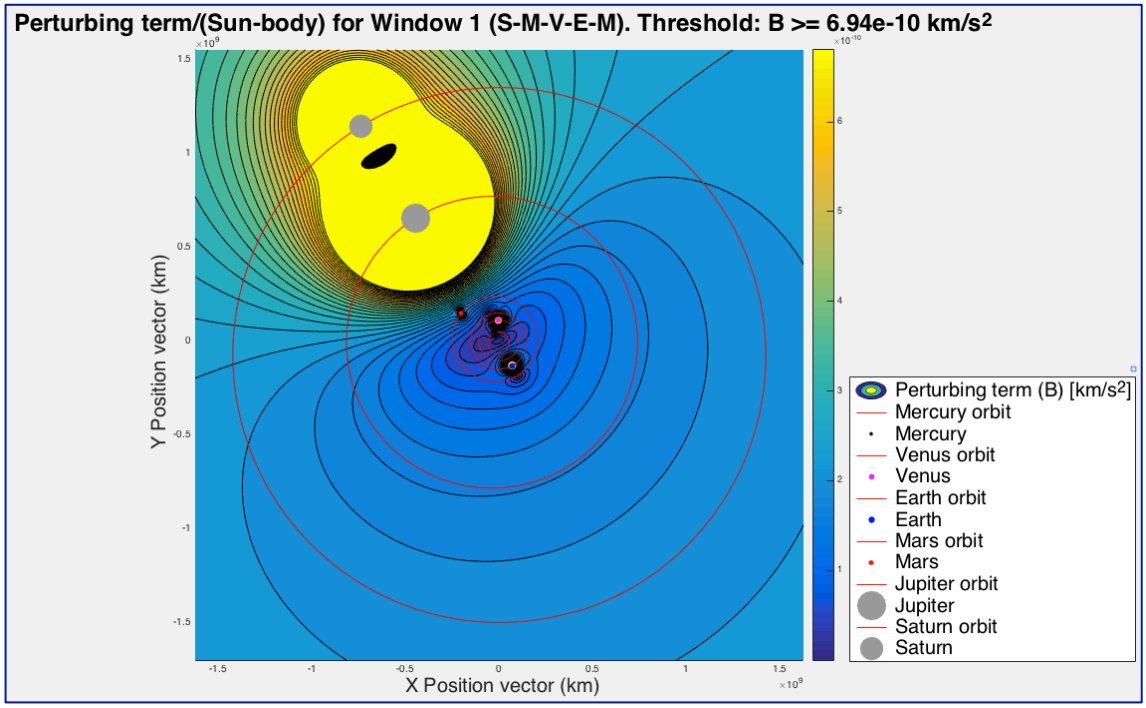

Fig. 6.13 2D Map for Window w + Planetary configuration scheme. Case of minimum distance alignment between Jupiter and Saturn (≈3.81 AU).

Since their orbits are fairly non-concentric, that minimum distance alignment will rarely happen. However, the disturbance circles are large enough for any simple alignment - or even when the planets are within an approach phase - to occur an overlap. **Figure** 6-14 magnifies again the picture to analyse it better.

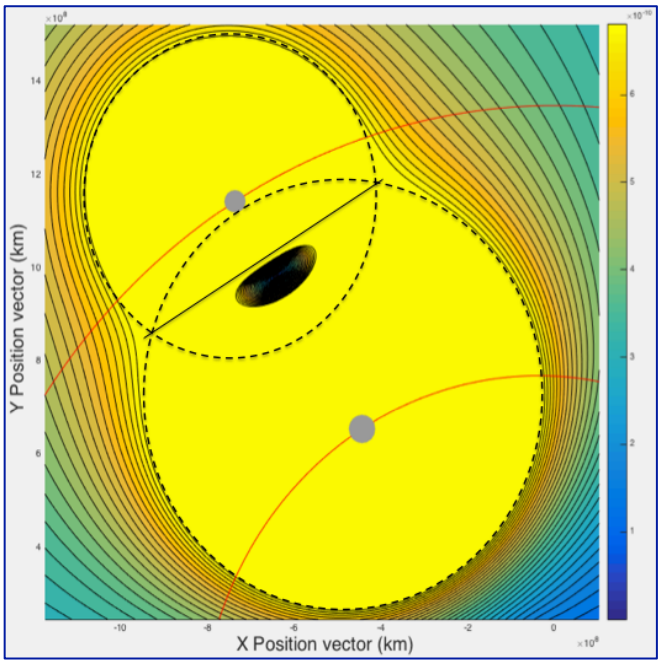

Fig. 6.14 Magnification of the intersection of PZs in the case of minimum distance alignment between Jupiter and Saturn.

The same discussed before regarding a not perfect symmetry due to discretisation reasons applies as well here. Moreover, the intersection in the auxiliary circumference sections produces two circular segments, leaving the ellipse clearly associated with the destructive interference zone in the segment corresponding to Jupiter That occurs because it is about 3.34 times more massive and, consequently, its disturbance zone is introduced much more into that of Saturn than does that of the latter in Jupiter (it can be observed that the auxiliary circumference of Jupiter reaches almost the position of Saturn in its orbit; remember that the size of the planets is symbolic).

Before further analysis, the preliminary conclusion is the same that for the case of Venus-Earth, *the circumferences of both planets overlap, giving rise to areas where interaction is constructive and areas where it is destructive*. **Figure 6-15** shows a picture analogous to the one represented in **Figure 6-10**.

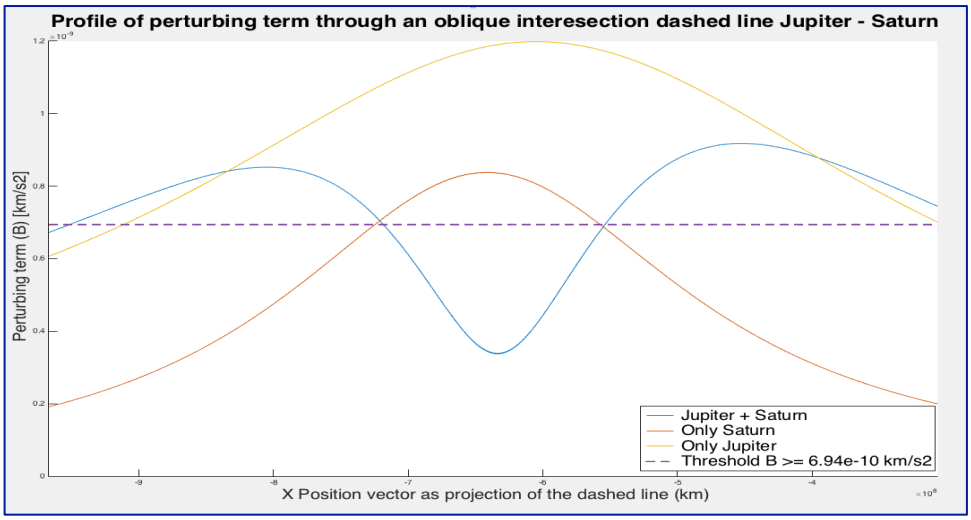

Fig. 6.15 Profile of perturbing term through a symmetric intersection (Jupiter – Saturn).

Similar conclusions can be achieved (see also **Figure 6-16**):

1) As with Venus-Earth, the combined profile reflects the synergy by having its disturbance value above that of the individual in the initial leg. Jupiter's profile surpasses it much earlier than Saturn's because the circular sector generated by the straight line joining the intersection points of the auxiliary circumferences is larger for Jupiter ($M_J \approx 3M_S$).

2) The individual profiles present a very similar behaviour, but exhibit an offset and different maximums (in value and in position after exceeding the threshold) because they correspond to different circle chords (both in length and in penetration inside the disk). **Figure 6-16** will help no note this.

3) In Saturn and combined profiles, the threshold is exceeded in almost coincident positions (upper and lower surpassing) because the length of the destructive interference zone is approximately the same of Saturn's chord.

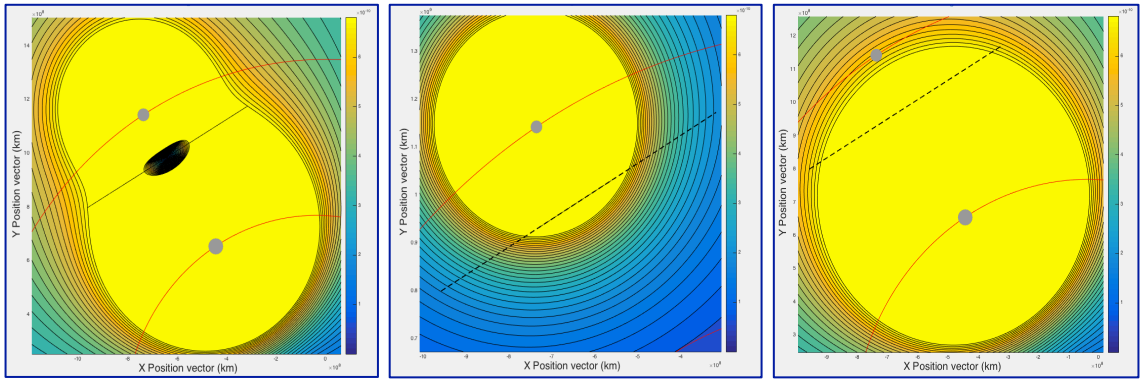

Fig. 6.16 Perturbation Zones: Left) Jupiter + Saturn[99]. Centre) Only Saturn (Jupiter removed). Right) Only Jupiter (Saturn removed).

6.2.3. Conclusions

Once the 2D mapping tool has been developed, now is the time to summarise and reach general conclusions from previous analyses to make these results useful:

- 2D maps show the disturbances of the acceleration field of a body – asteroid or spacecraft – relative to the Sun. Their objective is to serve as a "guide" to predict which celestial bodies to implement in interplanetary trajectory simulations
- Two antagonistic scenarios have been identified for both windows: - The generic one in which no planet is aligned with another and in which two of them are.
- The general scenario - the most probable, repeated and expected in time - gives as its main result that disturbance zones have a circular geometry (spherical in space) and are isolated from each other, so that at no point is the perturbative acceleration threshold exceeded by the action of more than one planet. In addition, it has been observed the Perturbation Zones conserve its size (radii)

On the other hand, regarding the particular scenario of alignment:

- Even at the point of maximum proximity, the disturbance zones of the Mercury-Venus and Earth-Mars systems never overlap. They behave to all intents and purposes as in the general case.
- For the Venus-Earth and Jupiter-Saturn systems, any alignment or approach phases produces an overlap that leads to areas where constructive interference and destructive interference alternatively occurs.
- In the case of overlapping, the circular sectors generated by the intersection of auxiliary circles show that the degree of penetration of one disturbance zone into another is a function of the masses of the bodies that create them. The greater the difference in masses, the less equitable the areas in these segments.

[99] The profile in **Figure 6-16** is obtained along the line that approximately divides symmetrically the elliptical destructive area that appears as a consequence of the interaction of PZs.

- The appearance of constructive/destructive zones – which always fall in the area of overlap between auxiliary circumferences – together with the analyses of the disturbance profile that crosses them, show that a collapsed system has both an ***advantage*** and a ***disadvantage***:

  + Unlike what would happen if there were no overlap – where the disc of one or the other body only would be crossed - it is possible to cross a more or less wide area in which the threshold is not exceeded, which opens the way to valid trajectory propagations with 2BP for the threshold that has been set as a reference for an electric propulsion.

  - In the constructive interference section, the value of the combined disturbance is greater than that of the individual disturbances in the absence of the homologous body. This leads us to think that crossing these sections would require the consideration of at least four celestial bodies.

And some additional conclusions (not the main ones) could be:

- The Moon produces a much smaller disturbance field than the Earth (in size and magnitude) but is large enough to include the Earth and its orbit in it

- In the propagation of trajectories of spacecraft leaving a GEO orbit, the Moon should be included, due to its significant perturbation relative to the Earth, its rapid orbital movement around the Earth and the fact that an electrically propelled spacecraft presents a low maximum acceleration capacity.

6.3. Spheres of Perturbation. Limitations of the tool

The tool/development that has been followed throughout this work has already been obtained. The 2D maps provide the necessary information to apply it to any interplanetary mission planned with BepiColombo or with another S/C that presents a similar propulsion capacity (representative). However, the performance of these previous analyses each time to delve into the behaviour of a given trajectory in the defined control volume is not the most desirable. In order to obtain an automated tool, the concept of ***Perturbing Sphere*** (***PS***), will be used, which is immediately described.

As mentioned previously, the algorithm that creates 2D maps generates a random planetary configuration each time it is executed. The analyses showed that in very few cases (and very infrequently) there is an overlap of two PS (or PZ, the first being more specific). In fact, in the vast majority of cases, *these PS are always isolated and exhibit the same sizes* (i.e., the same radii). This is probably the most useful result[100] of these analyses and what makes the tool to be automated (as will be demonstrated in the next chapter, when validating it).

[100] As often happens in science, “luck” is important. It is desirable to admit that he has also played in favour. Probably due to threshold obtained in **Chapter 5**, the PS (which are not small as they are more than an order of magnitude larger than any SOI) could have been even larger, which would have led to a multiple overlap problem. The problem would not be the overlap itself (since several bodies can be included in the equations of motion at the same time when propagating) but the difficulties that might have arisen in measure the value of the radii.

Table 6-1 shows the measured PS and the SOI for the same planets, to achieve an order of magnitude about their size.

Body	PS radius (10^6 km)	SOI radius (10^6 km)
Mercury	6.11	0.112
Venus	22.5	0.616
Earth	24.8	0.924
Moon	2.75	0.0661
Mars	8.87	0.576
Jupiter	506	48.2
Saturn	269	54.6

Table 6.1 Measured PS and references for average SOIs.

There are several relevant considerations regarding the measurement of the radii of PS to mention:

- In the 2D maps PS are visualized as circular zones, because all the planetary orbits have been projected in a plane, but they should be spheres in the space.
- They have been measured directly from the maps in general scenarios, since it is necessary to verify that they conserve their circular symmetry approximately intact.
- As the values can vary slightly depending on where the measures are taken since the map constitute a window of discrete coordinates, for safety the maximums will be taken, rounded up in the second significant figure.
- For Jupiter and Saturn, due to the fact that they are the ones that most clearly present zones of influence that are not perfectly circular, the largest radius will be taken as the radius (for safety's sake).

Limitations of the tool/procedure

There is a strong limitation based on the following facts:

- The disturbance threshold set was obtained as 1 % of the maximum accelerative capacity exhibited by BepiColombo, which is based on the fact that ESA has a 'margin philosophy' document to provide safety factors.
- In case the interplanetary trajectory to be applied was an asteroid rendezvous departing from Earth, the Lambert Arc[101] has the limitation that it makes use of an output speed (ΔV_{total} after applying a differential corrector) that will provide high orders of magnitude compared to what the low-thrust can give (i.e., technically it is high - thrust).

[101] When referring to Lambert Arc, what is meant to pinpoint is that the Arc provides the velocity vector differentially corrected at the departure to make it possible to reach the destination in the imposed time-of-flight (i.e, once defined the couple of dates departure/arrival). The idea of applying it to an asteroid rendezvous by means of incorporating the corresponding perturbations is to use that time-of-flight and the velocity at the departure as initial state and time of propagation, respectively using the general equations of motion (not the Arc itself, which is used for solving the 2-Body Boundary value problem).

Therefore, it would not be representative at all, since it would be necessary to propagate considering the thrust that can actually give the S/C (low-thrust). This is important because it would make the disturbances "fictitiously" affect less than they should. So, the application of the tool loses effectiveness (it will constitute a good starting point for future further developments that succeed the present work).

CHAPTER 7. VALIDATION OF 2D MAPS: USE OF BEPICOLOMBO EPHEMERIS

This last chapter will be devoted to the validation of the tool/procedure once it has been developed and is able to provide useful information for interplanetary trajectories. This will be done by simulating a real interplanetary ephemeris trajectory through the numerical integration of the equations of motion, using two special perturbation models: *Cowell* and *Encke*. The purpose of the process is to demonstrate that no significant difference is found in the motion of the spacecraft by propagating the initial ephemeris conditions including all the planets in the perturbing term or simply those indicated by the 2D maps (through the application of the PS concept developed some lines above).

7.1. Introduction

The 2D maps show the perturbed acceleration fields relative to Sun for two simulation windows, which contain all planets included in the control volume defined to represent the part of the Solar System of interest (see discussion in **Chapter 4** for more detail). In order to arrange the ideas and remember the *key points*, some of the main characteristics of the 2D maps are condensed:

- They are generated through a code that allows mapping a grid of certain dimensions and contains a discretisation scheme of the potential positions that a body (e.g, a spacecraft or an asteroid) can occupy along its trajectory around the Sun
- Each map specifically shows in such grid the magnitude contours of the perturbative acceleration vector sum of all the planets/satellites considered
- These contours are plotted in such a way that it is possible to distinguish which areas of the discretised planar space exert a disturbance in the body greater than a given fixed threshold, based on the propulsive characteristics of an electric spacecraft acting as a representative of modern missions: *BepiColombo*
- The geometry of these perturbative areas is circular (it is spherical in space)
- Each map is obtained by running the code, which leads to a random planetary configuration each time. Thus, the distribution of the planets is fictitious but not their perturbative spheres (which indeed remain constant in size)

Therefore, 2D Maps provide – for a given trajectory – information about where such perturbative acceleration threshold is exceeded and which celestial bodies are responsible, but they must be properly validated. Since the validation will require a well–known trajectory taken as a reference, *the best candidate is none other than the spacecraft used to compute the perturbative acceleration threshold: BepiColombo.* The logical steps to follow are:

1) The real ephemeris trajectory data are imported from the HORIZONS web - interface (https://ssd.jpl.nasa.gov/horizons.cgi#top)

2) A propagation of such trajectory will be carried out in *MatLab* using its initial conditions (position and velocity) and numerically integrating the N-body motion equation. First, when using 2BP a large discrepancy should be noticed

3) Information provided by 2D maps must allow propagating a more accurate trajectory (i.e., decrease in the discrepancy) by adding the appropriate bodies to the perturbing term[102]

Finally, the specific objectives to be covered are:

- Qualitative analysis of Bepicolombo's ephemeris trajectory based on the information provided by 2D maps (application of PS concept)
- Trajectory propagation using 2BP, based on initial BepiColombo conditions provided by ephemeris data. Discrepancy analysis (called "Error Function")
- Comparison of the Error Function with that obtained when considering all the planets of the solar system (so, using full NBP)
- Trajectory propagation considering in the perturbative term only those planets whose perturbative spheres have been crossed at some point during the simulation[103] (***Validation of 2D maps***)
- Trajectory propagation considering those planets whose perturbative sphere has been crossed but only at the integration step when it happens (***Validation of the dynamic trajectory propagator algorithm***)

7.2. Qualitative description of BepiColombo's ephemeris trajectory. 2D Maps output

7.2.1. Actual trajectory imported from HORIZONS

The key points for the representation of the actual trajectory designed for BepiColombo's mission by the navigation teams of JPL are:

[102] Remember that the purpose of this thesis is not to emulate ephemerides trajectories with high accuracy, but to compare the observed discrepancies between the propagation of the full NBP and that of the partial NBP (including only the relevant celestial bodies in the perturbing term). This is so provided that the simulated trajectory with all bodies presents a reasonable fit with that of the ephemeris, since only in this way can it be guaranteed to pass through approximately the same areas of space (which is crucial, since the concept of PS that are crossed is applied knowing a true trajectory, as will be seen in this chapter).

[103] Such information is directly gathered from the previous qualitative analysis of BepiColombo's ephemeris trajectory.

- BepiColombo's ephemeris trajectory is plotted from **2018 October 21th to 2025 November 2nd**[104] with a time step of 1 terrestrial day (24 h) (2570 days simulated)
- The Sun and those planets contained in simulation **Window 1** (Mercury, Venus, Earth and Mars) are included in the animation for the same dates and time step
- All trajectories are referenced to SSB. Thus, no coordinate transformation is needed to properly visualize the relative motion between the S/C and the celestial bodies
- The S/C motion is represented by a trace[105] that continuously changes colour to follow the instantaneous position. The motion of the planets appears in white/blue alternatively for each new orbit to allow differentiation. For Mars and the Earth such transition is very well visualised – probably due to the lesser effect of gravitational deviation exerted by the Sun – but worse visualised for Venus and really deficient for Mercury (the Sun induces a large deviation between consecutive orbits)
- The fly-by at Earth is clearly identified. The first one at Venus is not well seen but easy to intuitively guess. Venus second flyby and the six of Mercury cannot be identified using the MatLab animation function. (see **Figure 7-1**)

Distance history

A record of distances is acquired from BepiColombo spacecraft to each planet to detect when a crossing of each perturbative sphere occurs (if it does) along its actual trajectory. Results are shown in **Figure 7-2**.

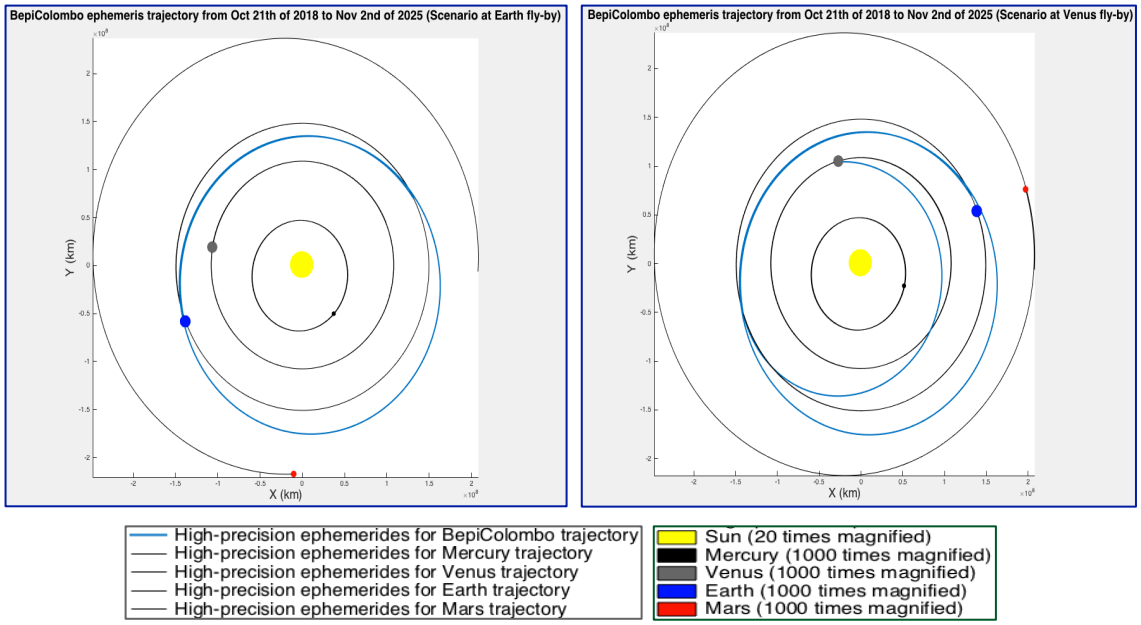

Fig. 7.1 BepiColombo ephemeris trajectory from Oct 21st 2018 to Nov 2nd 2025. Left) Scenario at Earth's flyby (in the animation 00:36 s). Right) Scenario at Venus first flyby (in the animation 00:48 s).

[104] This is due the warning issued by the HORIZONS platform that no data is available outside of those dates).

[105] To visualise the animated plot created in the MatLab function that can be played if desired.

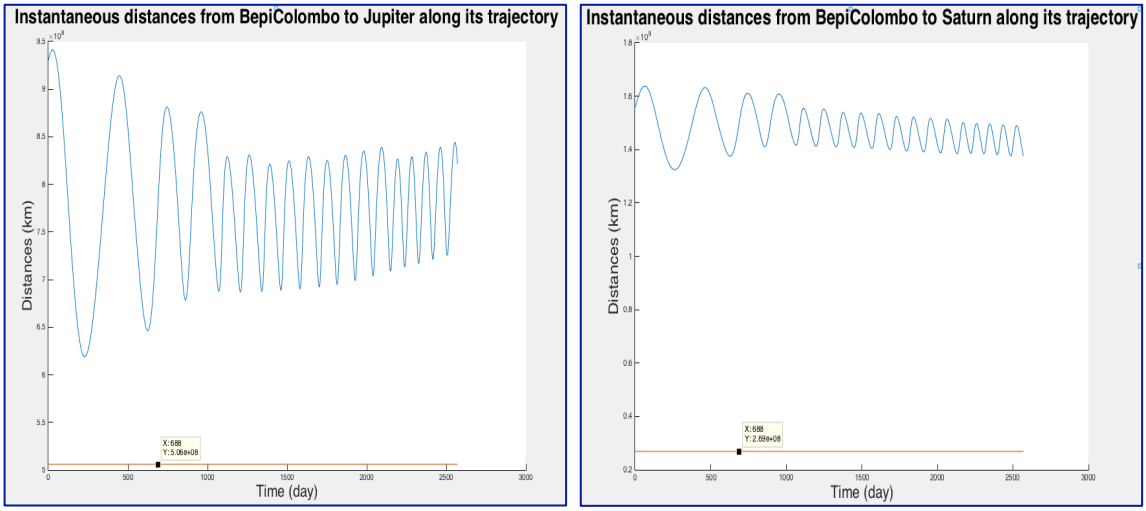

Fig. 7.2 Instantaneous distances from BepiColombo to: Left) Jupiter. Right) Saturn. The red straight line corresponds to the radii of their respective PZs.

It can be observed that the S/C is really far from their Perturbation Zones at every time of the path. Obviously, in the case of Jupiter the spacecraft is much closer than in the case of Saturn. **Figure** 7-3 shows the analogous pictures for Mercury and Venus.

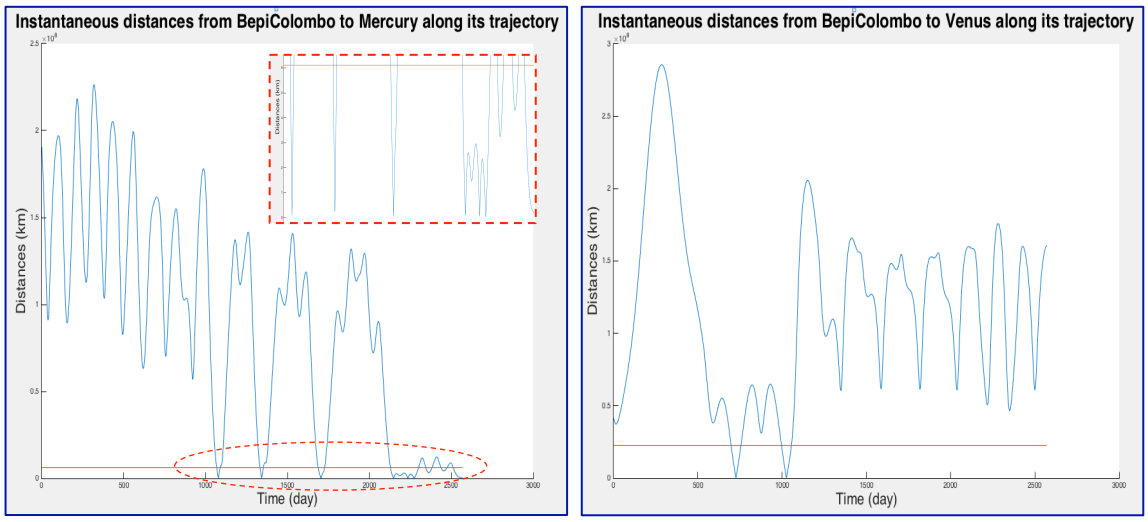

Fig. 7.3 Instantaneous distances from BepiColombo to: Left) Mercury. Right) Venus. The red straight line corresponds to the radii of their respective PZs.

Six bands can be identified that cross the surface of the sphere of perturbation of Mercury but only the first four[106] correspond to flybys, as the magnified capture shows, since the minimums have a similar order of magnitude (in terms of distance). For Venus, only two crossings take place, which are associated to the two planned flybys for the mission. **Figure** 7-4 shows the remaining planets: Earth and Mars.

[106] The fourth band contains 3 consecutive flybys, remaining the S/C inside the perturbation sphere at every time; On the other hand, the last crossing just means the final arrival to the destination.

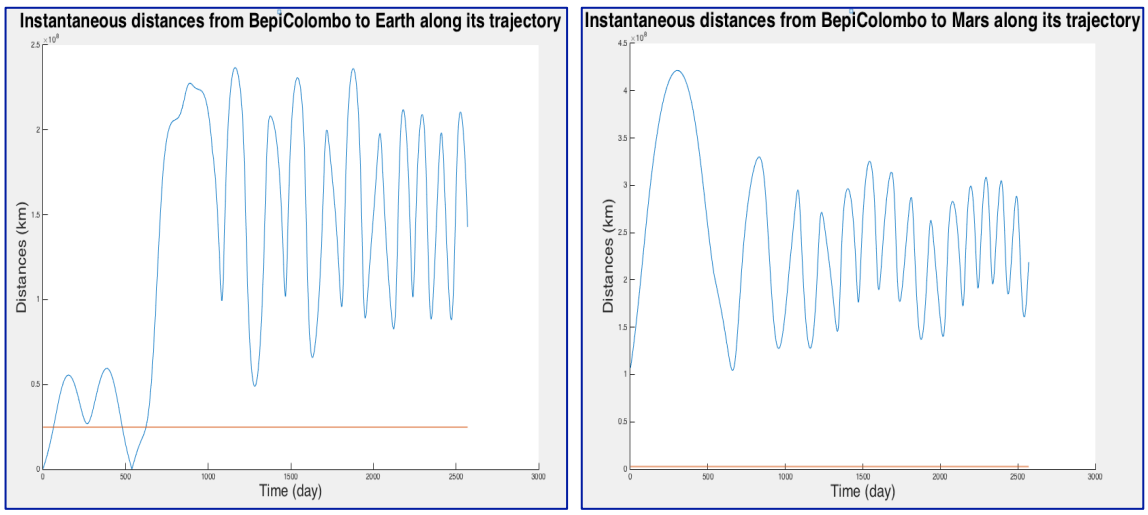

Fig. 7.4 Instantaneous distances from BepiColombo to: Left) Earth. Right) Mars.

For Earth, only one band that crosses the threshold is identified In this case, it coincides with the only flyby that takes place (there is another band that almost crosses the minimum). The first crossing corresponds to the departure from Earth. For Mars – as with Jupiter and Saturn – the S/C is very far from its perturbation zone at every moment of its trajectory (but logically, considerably closer than that of those outer planets).

Statistics (times and lengths covered)

During the previous simulation, the algorithm stores both times spent by the S/C within each Perturbation Sphere (PS) and the distances travelled during these intervals. This is necessary because, in practical terms, here is where the tool/procedure "*pulls the muscle*" (where its application takes place). **Table 7-1** and **Table 7-2** show the results for both ***times*** and ***lengths***. Some comments:

- Data are recorded for each planet in individual *path sections* (it corresponds to the section of the S/C trajectory where it remains within a given PS; Mercury has seven path sections – the last being without escape from its sphere since it constitutes the arrival – while Venus and Earth have two, the latter including the departure)
- What is called "*length covered*" is actually the variation of the position vector module, as it is not possible to know the mathematical function related to the BepiColombo's traced trajectory, so it is just an approximation (see **Figure 7-5**)

Planet \ Path section	Mercury	Venus	Earth	Mars	Jupiter	Saturn	
1: Start–Finish (day)	1069-1090	698-756	1-64	-	-	-	
2: Start–Finish (day)	1333-1354	997-1056	481-624	-	-	-	
3: Start–Finish (day)	1688-1727	-	-	-	-	-	
4: Start–Finish (day)	2129-2301	-	-	-	-	-	
5: Start–Finish (day)	2345-2383	-	-	-	-	-	
6: Start–Finish (day)	2438-2474	-	-	-	-	-	
7: Start–Finish (day)	2513-2570	-	-	-	-	-	*Total*
Time inside PZ (days)	**385**	**117**	**206**	**0**	**0**	**0**	**708**
% Total trajectory	**14.98**	**4.55**	**8.02**	**0**	**0**	**0**	**27.6**

Table 7.1 Number of days that the S/C remains within each PS (broken down into the several path sections and the total accumulated).

Planet \ Path section	Mercury	Venus	Earth	Mars	Jupiter	Saturn	
1: Start–Finish (day)	$9.19 \cdot 10^{7}$	$1.60 \cdot 10^{8}$	$1.66 \cdot 10^{8}$	-	-	-	
2: Start–Finish (day)	$9.04 \cdot 10^{7}$	$1.42 \cdot 10^{8}$	$2.49 \cdot 10^{8}$	-	-	-	
3: Start–Finish (day)	$1.25 \cdot 10^{8}$	-	-	-	-	-	
4: Start–Finish (day)	$1.30 \cdot 10^{8}$	-	-	-	-	-	
5: Start–Finish (day)	$1.06 \cdot 10^{8}$	-	-	-	-	-	
6: Start–Finish (day)	$1.02 \cdot 10^{8}$	-	-	-	-	-	
7: Start–Finish (day)	$1.12 \cdot 10^{8}$	-	-	-	-	-	*Total*
Distance inside PZ (km)	**$7.57 \cdot 10^{8}$**	**$3.02 \cdot 10^{8}$**	**$4.15 \cdot 10^{8}$**	**0**	**0**	**0**	**$1.47 \cdot 10^{9}$**

Table 7.2 Length covered by the S/C within each PS (broken down into the several path sections and the total accumulated).

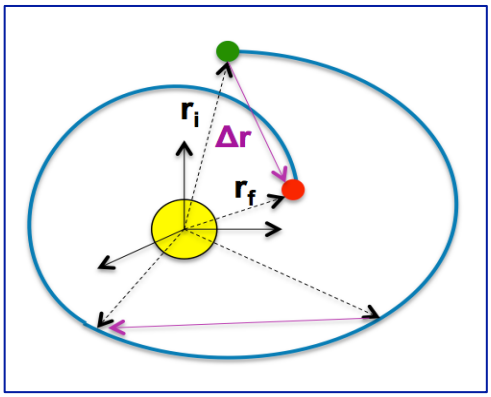

Fig. 7.5 Relationship between relative vector position and distance travelled. The green point represents the entry into a PS. The red point corresponds the exit from a PS. The trajectory followed is drawn in blue.

According to **Figure 7-5**, the Δs covered by the S/C between the entry and exit positions of a Perturbative Sphere is larger than Δ**r**, however since Δr tends to Δs as they get closer and the largest path section is 172 days out of a total of 2570 (spent in Mercury's PS) this is an acceptable approximation. Anyway, the magnitude of the relative position vector Δr in a generic given trajectory may differ greatly from the actual length travelled Δs as seen.

Note: - In **Table 7-2**, the % of length covered within each PS over total trajectory is omitted, as accurately approximate Δr_{tot} to Δs_{tot} would require a time step << 1 day (the computational cost would be greatly increased, due to the large amount of ephemeris data needed). On the other hand, results published in **Table 7-1** will be used for later analyses and discussions.

7.2.2. Validation of the planar approach used in 2D Maps

Once the ephemeris trajectory has been described, some discussions will be made to get useful conclusions that can help to the application of the maps. They will based exclusively on the time information presented in **Table 7-1**, as it is the most accurate and the most important (deviations in a trajectory occur as a consequence of the time that $a_{perturbative}$ acts on a S/C). The following facts observed in the previous analyses will be related to the information that 2D maps provided.

1) BepiColombo is never within two or more Perturbation Spheres at the same time *(***Figures** **6-7** and **6-8** *have been recovered in support)*

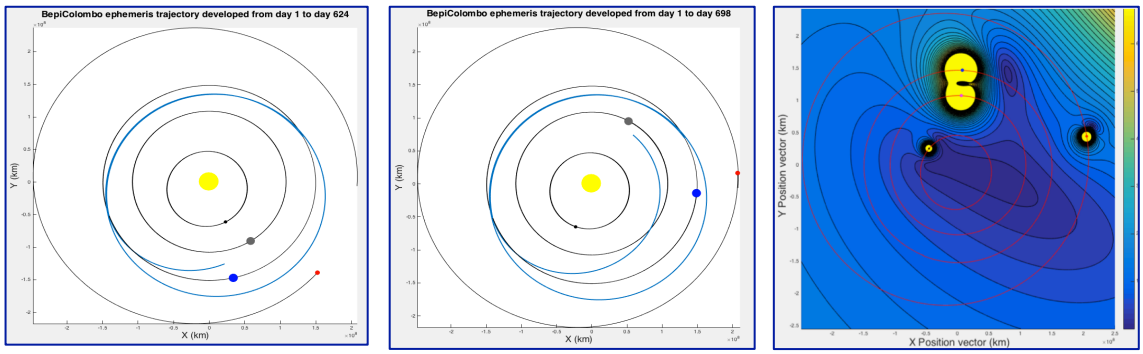

Fig. 7.6 Transition PZ Earth ➔ PZ Venus (74 days)

After Earth's flyby, the S/C leaves its PZ when both planets are clearly misaligned (picture on the left) being even more so when it reaches Venus PZ (centre). The information of the 2D map (right) showed that, indeed, only in the alignments – which occur every 592 days – or close to them, there could be an overlap in the unlikely event that the S/C was between the planets at that time).

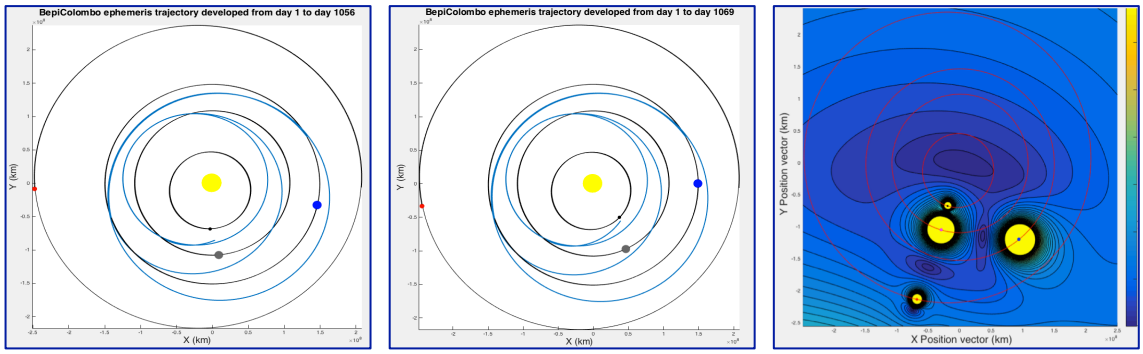

Fig. 7.7 Transition PZ Venus ➔ PZ Mercury (13 days)

After Venus 2nd flyby (left) the S/C leaves its PZ when both planets are slightly misaligned, though not far from it. Those just 13 days seem to show that an overlap was about to occur, but flyby events make the trajectory change very fast (approaching Mercury and moving away from Venus). So, in these cases, time can be deceptive (on day 1069 – centre – the S/C is already far from Venus PZ). This fact also supports the information of 2D map, which indicated that there could never be an overlap of the PSs of these planets (right).

2) Perturbation Spheres of Mars, Jupiter and Saturn are never crossed

- For the trajectory designed for BepiColombo, the consideration of these planets in the perturbative acceleration term of the equations of motion can be neglected and significant consequences should not be expected

- Nothing can be added from the point of view of 2D maps to this fact, since they simply show the magnitude contours of the perturbative acceleration vector sum for a discretised grid of potential positions that the S/C can occupy in its motion with respect to the Sun. Therefore, *which PSs are crossed and which do not depend on the particular trajectory itself*.

- There is a limitation[107] based on the fact that, when the S/C crosses areas where the perturbative acceleration threshold it is not exceeded, the disturbance term need not be considered in the propagation, because theoretically the S/C itself would be the one who cancels those perturbations it (it would have sufficient propulsive capacity to cope with it). However, the simulated trajectory for producing the ephemeris data probably took into account all the bodies and not that cancelation capability.

3) Approximately one third of the designed trajectory takes place with the S/C within a PZ (708 days ≅ 27.6 % of the time-of-flight)

- It indicates that to consider the proper bodies in the perturbing term when integrating the N-body equation of motion should be absolutely crucial (to simulate)

- Apparently, the time accumulated by the S/C within each PS could mean that the order of priority of the perturbative bodies should be considered (the S/C spends almost twice as much time inside $PS_{Mercury}$ as within PS_{Earth} and about the same proportion for PS_{Earth} / PS_{Venus}). However, such an arrangement would be incorrect. Mercury alone would not improve a trajectory propagated through 2BP more than it would by introducing only the Earth or only Venus. Due to the peculiarity of existing flybys in the trajectory, adding only Mercury would be useless without the other two, (due to the need to divert the S/C first to allow approach it)

From this discussion, two overall conclusions can be reached, which are:

1) Propagating initial conditions of BepiColombo's ephemeris (position and velocity) by adding three bodies to the perturbing term (Mercury, Venus and Earth) should lead to an accurate replica with small deviations (small discrepancy).

2) Since the S/C is never within two PZs at the same time, no significant changes between using the static or the dynamic propagation should be noticed.

Having assumed these results now is the time to apply the tool developed in *Chapter 6* (through the Perturbation Sphere concept) and demonstrate that, in addition, the planar projection approach carried out is more than acceptable and should enable the maps to provide reliable results. To do this, the PS radii will be corrected using the 3D trajectory of the ephemeris. It is necessary to remember that they were found by examining the limits within which the perturbative threshold was exceeded. Thus, by calculating the instantaneous value of perturbation magnitude along the entire trajectory of BepiColombo, it will be possible to verify if the threshold overrun occurs at the exact distances from each planet shown on the 2D maps, or before or after. The steps to follow are:

[107] Such limitation is related to what was drawn in **Figure 4-2** at the Section 4.1 in Chapter 4.

1) Correction (necessary to later apply a *dynamic propagator algorithm*, which will need to differentiate which planet is responsible for the exceeding the threshold at any given time)

2) Distance and time recalculations

3) Verify the degree of success in predicting 2D maps

The results are shown in **Figure 7-8**. It represents the magnitude of the instantaneous vector sum of the perturbing accelerations along the BepiColombo's ephemeris trajectory. If the current radii provided by the 2D maps are exact (or close to being exact) what should be seen is that the threshold is reached only at (or near) those values. The "current" PS surfaces were crossed at the times indicated by **Table 7-1**. The diagram is cut at the top to focus on the threshold crossing (orange line).

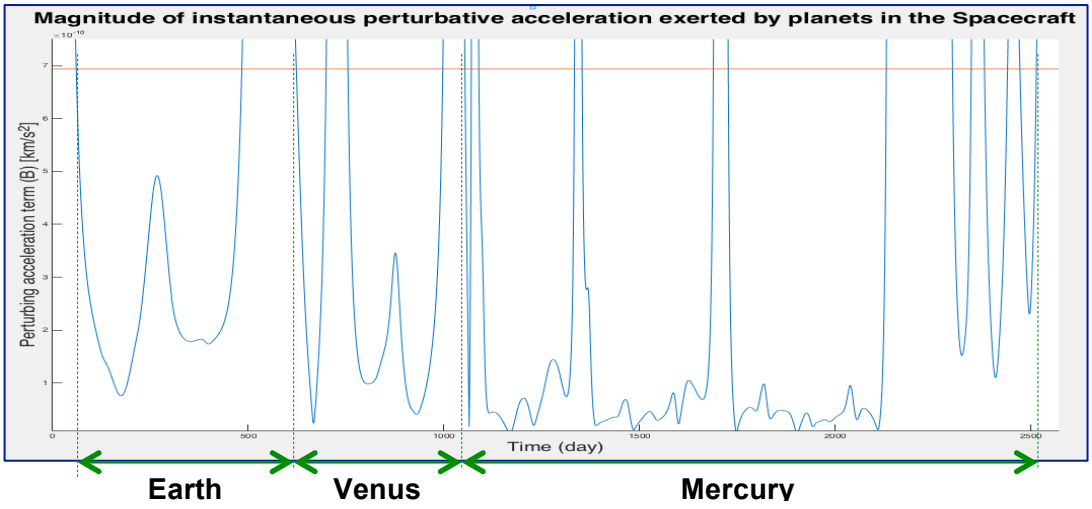

Fig. 7.8 Magnitude of the instantaneous vector sum of perturbing accelerations exerted by the celestial bodies in BepiColombo spacecraft. The orange line is the threshold.

This diagram is exactly the inverse of those of distances to each planet in **Figures 7-2**, **7-3** and **7-4**. It is like the "fingerprint" of each dominant planet; e.g, for the Earth the form "M" in its distance plot becomes "W" here, the extended "M" for Venus becomes a extended "W" here, etc.). Each section has been qualitatively delimited to differentiate when the S/C is under the unique influence of the Perturbative Sphere of each planet (exceeding the threshold both upper and lower). Thus, when the next section begins, the S/C has already left the PS of the previous planet (after its last flyby) and approaches the next one, without ever re-entering the previous PS). **Table 7-3** finally shows the updated times that BepiColombo spent within each actual PS, which is simply the times where the threshold is exceeded (upper and lower).

Compared to those obtained in **Table 7-1**, it is clear that, the time spent by the S/C within each PS has suffered a regression, both when entering (delaying that day) and when leaving (advancing that day). This involution is very small and represents a total of 26 days less (≈ -1%) within some PS (therefore, reducing a little the time in which the perturbative term does not need to be considered when propagating a trajectory).

In addition, each PS lost a number of days proportional to its original value (15 for Mercury, which is roughly the double that the 7 lost by the Earth, which at the same time is about double the 4 lost by Venus) thus *the spheres are slightly smaller but still proportional*. The actual PS have been corrected and exhibit a high degree of agreement with the original ones measured directly from 2D maps in *Chapter 6*. **Table 7-4** shows the comparison.

Planet \ Path section	**Mercury**	**Venus**	**Earth**	**Mars**	**Jupiter**	**Saturn**	
1: Start–Finish (day)	1070-1089	699-755	1-61	-	-	-	
2: Start–Finish (day)	1333-1353	997-1054	484-623	-	-	-	
3: Start–Finish (day)	1688-1726	-	-	-	-	-	
4: Start–Finish (day)	2130-2300	-	-	-	-	-	
5: Start–Finish (day)	2347-2381	-	-	-	-	-	
6: Start–Finish (day)	2440-2472	-	-	-	-	-	
7: Start–Finish (day)	2514-2570	-	-	-	-	-	*Total*
Time inside PZ (days)	**370**	**113**	**199**	**0**	**0**	**0**	**682**
% Total trajectory	**14.40**	**4.40**	**7.74**	**0**	**0**	**0**	**26.5**

Table 7.3 Number of days that the S/C remains within each PS (broken down into the several path sections and the total accumulated). Updated

As with the times, the updated radii are fairly close to the original values, which confirms the quality of the planar projection for constructing the 2D maps. Since the S/C did not cross the Perturbative Spheres of *Mars*, *Jupiter* and *Saturn*, it has not been possible to adjust their original radii (although they are expected to vary in the similar fashion). The Moon, on the other hand, was preferred not to be included in this qualitative analysis to avoid confusion when representing the Earth in the animated plot. It will be considered in the next sections, as it is really close to the Earth, so it is expected to exert a contribution. Taking into account how small the PS radii variation is, we choose to keep *the old radii* in order to be conservative and add a safety factor.

Body	PS radius (10^6 km)	Updated PS radius (10^6 km)
Mercury	6.11	5.72
Venus	22.5	21.6
Earth	24.8	23.5
Moon	2.75	2.75
Mars	8.87	8.87
Jupiter	506	506
Saturn	269	269

Table 7.4 Left) Old PS radii. Right) Updated PS radii. It can be observed the high degree of agreement, which shows that the planar projection approach is valid.

To finish this section, the conclusion that it is worth to pinpoint is that, the qualitative analyses carried out for BepiColombo's ephemeris trajectory therefore showed that, being aware that it is still early to accept a successful performance of 2D maps (which will be tested basically in the following sections) it can be confirmed that *the planar projection of the solar system in the ecliptic was a very good approximation.*

7.3. Trajectory propagation using 2BP model

In order to promote a first simulation and analyse the first results (which will be useful later) the initial conditions foreseen of BepiColombo's ephemeris will be integrated using 2BP. As we have already explained in detail both the management of ephemeris data and the proper transformation between reference frames (from SSB to that relative to the body) and also the correct introduction and updating of the dynamic planetary state vectors, here we will present only the results and analyses/discussions derived from (thus avoiding the technical details – that can be consulted at **Chapter 2** and **Chapter 3** – unless strictly necessary for the understanding of the rest, as will be the case with later flybys modelling).

Figure 7-9 shows the propagation results for a simulation time equal to the real mission (2570 days) and a time step of 6 h. In the second picture it can be seen that, the trajectory provided by 2BP (orange) does not fit at all with the real trajectory designed for BepiColombo's ephemeris (blue). In the simulated trajectory, the S/C continues orbiting indefinitely around the Sun since no other body induces a deviation (there are no perturbations or flybys)

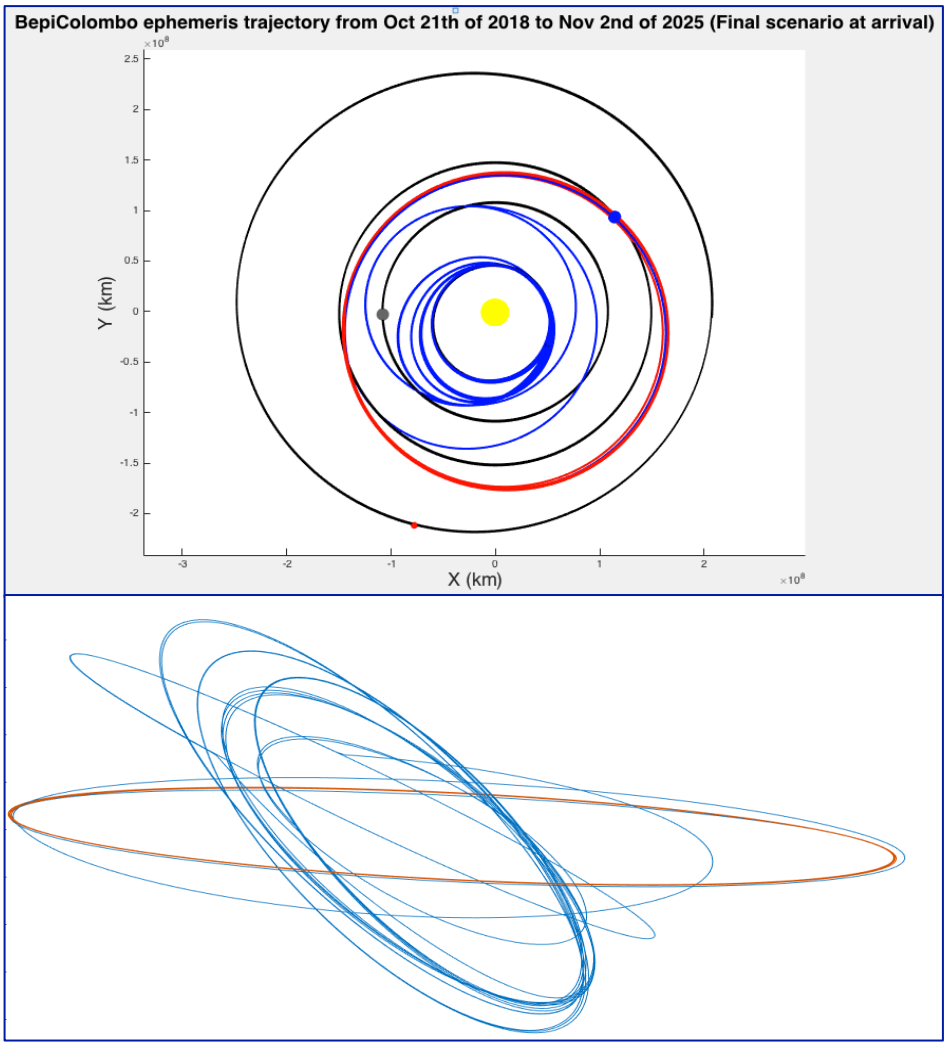

Fig. 7.9 Trajectory performed by BepiColombo (Orange, simulated 2BP ; Blue, HORIZONS ephemeris).

Since the actual representation of the trajectories followed does not provide an accurate idea of the magnitude with which they differ, **Figure 7-10** shows the discrepancy between them. It is observed to be very high: 2.595e+08 km (1.734 AU) for the maximum and 1.322e+08 (0.884 AU) on average. The fact that there is a relatively much smaller error in the first section is due to the fact that real BepiColombo has not yet performed its first flyby, so its path still remains similar to that of a solar orbit. Once the S/C reaches Earth (which occurs around day 537) the ephemeris trajectory begins to divert, while that of 2BP does not. The conclusion is the expected one: the *2BP-simulated trajectory exhibits a very poor fidelity*.

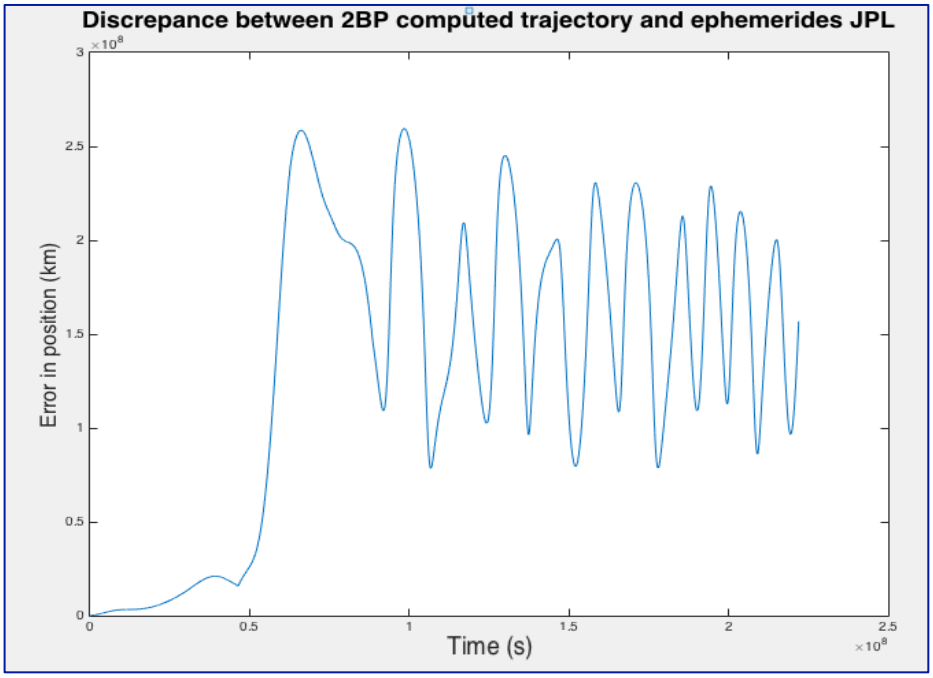

Fig. 7.10 Discrepancy between the computed 2BP trajectory and ephemeris (Error Function).

7.4. Trajectory propagation using the complete NBP model

The complete N-body problem equations of motion (i.e, Equation 2-19) will be numerically integrated to propagate BepiColombo's ephemeris initial conditions (therefore, including ***all the planets of the solar system + Moon***).

7.4.1. Cowell implementation software

The objective is to simulate, as accurately as possible, the BepiColombo's ephemeris trajectory downloaded from HORIZONS. This will allow us to study later how they affect the disturbances produced by the bodies of the solar system in such trajectory. Some preliminary considerations to be taken are:

1) To download ephemeris data of the *bodies* and the *S/C* (position and velocity) with a **step size of 6 h** (instead of 24 h as in the qualitative analysis). It is based on the fact that, when motion is near a large attracting body, Cowell's method needs to take small integration steps.

2) To propagate the initial conditions of the S/C only until the **first flyby on Mercury** (***27000 h = 1125 days***). For the analyses of perturbations it is sufficient, since it implies that the spacecraft suffers the necessary deviations that lead to cross each one of the respective SOI towards the destination.

3) The flybys are susceptible of being modelled through two possible approaches:

3.1) Geometrically*, by means of applying the* ***link-conic approximation****, which is based on assuming the SOI of the planet to have zero radius and, consequently, considering such gravity assist manoeuvre instantaneous* [40].

Advantage: - Simple to implement and acceptable on the scale of the solar system for fast swing-bys.

Drawback: - It is not completely accurate since it introduces a discontinuity in the velocity vector but not in the position vector

3.2) Numerically, by means of fully propagating the S/C trajectory with the N-body differential equation of motion. The key is to change the central dominant gravitational body (the Sun) by the planet on which the flyby takes place. That shift is carried out when the S/C crosses the SOI of such planet. Therefore, the swing-by is treated as an isolated stage and then linked to the incoming and out coming parts (if known) of the trajectory, by propagating backwards and forwards in time from the pericentre to both sides of the SOI, while satisfying some matching conditions [41], [42].

Advantage: - Much more accurate (close to dynamic behaviour of real physics)

Drawback: - Drift in the numerical integrator. Depend on the accuracy of the model (in Cowell's, when motion is near a large attracting body, smaller integration steps are needed: increasing accumulative round off error)

Between the two options, **numerical integration is preferred** for the sake of precision. The modelling strategy for each flyby consists of following stages:

1st) When the algorithm detects that the S/C crosses the SOI of a major body (i.e, a planet) the original heliocentric propagation is paused to call a function that models the flyby.

2st) Said function will characterise the hyperbolic orbit that physically describes the gravity–assisted manoeuvre and compute its period (the time spent by the S/C within the SOI). Then:

i) It will propagate backwards in time the state vector (position and velocity) that the S/C exhibits at its minimum distance from the major body – pericentre of the hyperbola – up to the SOI's entry point (which corresponds to the last propagation step computed at 1st).

ii) It will propagate forwards in time the same state vector up to the SOI's exit point (which is defined at the time step when the distance of the S/C to the planet exceeds its SOI)

iii) It will return the control to the main algorithm as to continue with the heliocentric propagation of the until a new SOI is penetrated

Once all of above is clear, a question can come up: *what about the pericentre of the hyperbola to simulate first flyby on Earth*? Such a question will be answered after discussing the picture in **Figure 7-11**, which is that of **Figure 7-8** but without imposing the Y-axis limitation (value of the perturbative magnitude) to a value close to the perturbative acceleration threshold.

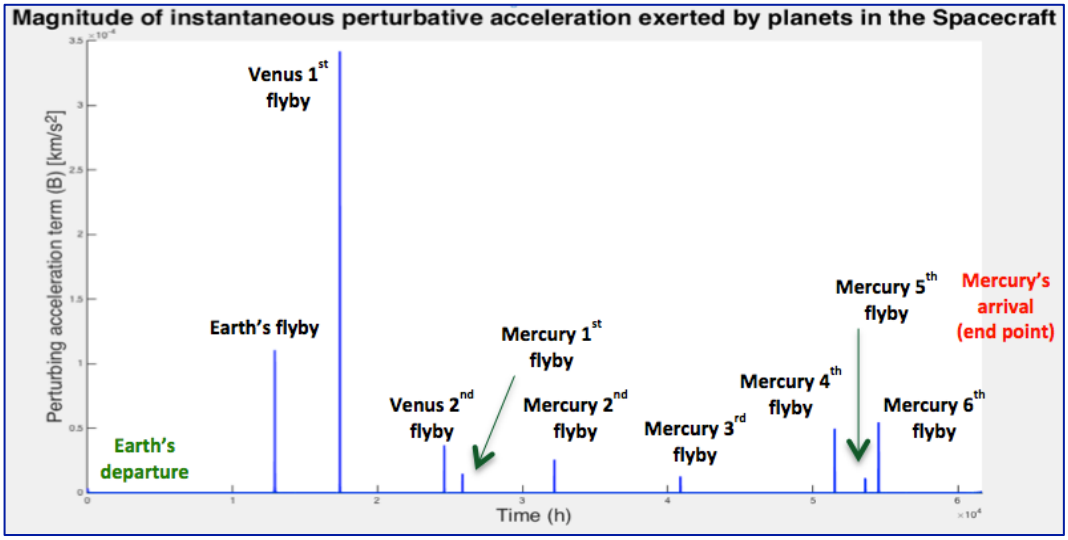

Fig. 7.11 Magnitude of instantaneous vector sum of perturbing accelerations exerted on BepiColombo's (Original size, without magnification).

This plot, despite of its anecdotic similarity to a "*chemical atomic spectrum*" is actually useful to identify each planetary flyby, since each "discrete line" corresponds to a minimum distance. These "lines" are not such, rather narrow bands that MatLab shows in that form due to the high perturbing values along a large time scale (which start at the time where a SOI is crossed).

Note: - The arrival to Mercury has no associated "peak", as the distance to the planet at the last ephemeris point is 1.8e+05 km.

If the propagation of the trajectory from the initial ephemeris conditions were sufficient accurate, the state vector obtained at the epoch when the ephemeris show the first crossing of the Earth's SOI would exhibit a low discrepancy with it. Therefore, the state vector at the pericentre of the hyperbola could have been computed through the orbital parameters of the latter state before crossing the SOI and then optimised to satisfy matching conditions in the SOI itself. Unfortunately, that point is very far from the limits of the Earth's SOI, as there is a significant drift caused by both the numerical integrator used (a Runge-Kutta – Fehlberg scheme of orders 4,5) and by the gravity perturbation model (Cowell) which leads to severely affect the time and cumulative error due to the round off[108]. The solution consists of *using the ephemeris state vectors at the epochs when the planet's SOI is crossed and at the minimum distance*, respectively.

Flyby modelling. Characterisation of the hyperbola

The dynamics of each gravity assist will be simulated in a planet–centred system, the S/C being subject to its gravitational attraction while it is within its SOI and the perturbation comes from the Sun (its initial and final points will be converted again into the heliocentric frame to join the section with the beginning and to continue with the heliocentric propagation, respectively). Hence, the equation of motion associated to these legs has the following shape:

[108] More on this later.

$$\ddot{\boldsymbol{r}}_{12} = \frac{d^2\boldsymbol{r_{12}}}{dt^2} = -\frac{\mu_1}{r_{12}^3}\boldsymbol{r}_{12}^3 - \mu_3\left(\frac{\boldsymbol{r_{32}}}{r_{32}^3} - \frac{\boldsymbol{r_{31}}}{r_{31}^3}\right) \quad \textbf{(7-1)}$$

Where:

$\mathbf{r_{12}}$ is the spacecraft position vector wrt the major body [km]

$\mathbf{r_{31}}$ is the major body position vector wrt the Sun [km]

$\mathbf{r_{32}}$ is the spacecraft position vector wrt the Sun [km]

μ_1 is the major body gravitational parameter [$km^3.kg^{-1}.s^{-2}$]

μ_3 is the Sun gravitational parameter [$km^3.kg^{-1}.s^{-2}$]

Other perturbations could be considered but are negligible for the time that the hyperbolic orbit lasts (it has been verified). The flyby manoeuvre starts at the pericentre (minimum distance to the planet) and will be obtained by propagation backwards and forwards – respectively – a time interval, which must be computed first (orbit period). Important considerations are:

- The time is initialised on 0 seconds (the t_{span} needs to be set through T)
- The state vector is known at the pericentre from the ephemeris[109]
- Since the propagation takes place in a planet–centred system, the state vector needs to be properly expressed in it (remember that ephemeris is in SSB)

The hyperbolic orbit is characterised by two branches towards which the trajectory tends asymptotically, which leads to the concept of the hyperbolic excess speed. It consists of a residual speed coming from a release or absorption of energy – depending on whether the body is gaining or losing momentum – thus allowing an accelerated manoeuvre *(i.e, increasing or decreasing the speed of the S/C and changing its direction an angle close to the angle between the asymptotes).*

Since the *specific mechanical energy* remains constant along an orbit, the energy equation can be written between two points, for instance, the pericentre and a mathematic point placed at an infinite distance from the major body:

$$\varepsilon = \frac{v_p^2}{2} - \frac{\mu}{r_p} = \frac{v_\infty^2}{2} - \frac{\mu}{r_\infty} \quad \textbf{(7-2)}$$

Writing the conservation of angular momentum between two arbitrary points, it is possible to reach this general expression, which is valid for all conic orbits:

$$\varepsilon = -\frac{\mu}{2a} \quad \textbf{(7-3)}$$

Combining both expressions it is possible to reach a new one that tells how to calculate the semi-major axis for the hyperbolic orbit (since the last term in Equation 7-3 is zero):

[109] It is the state vector corresponding to the minimum distance to the planet. The associated time has been determined from the qualitative analysis made earlier.

$$a = -\frac{\mu}{v_\infty^2} \tag{7-4}$$

Given that it is a mathematical concept, the hyperbolic excess speed will be only exactly reached at the infinite, but an acceptable approximation would be just to take as that speed the one that the S/C has when crossing the SOI (*incoming velocity modulus*). Therefore:

$$v_\infty \approx v_{inc} \approx \left|\vec{v}_{Bepi_{SSB_{eph}}}(t_{SOI})\right| \tag{7-5}$$

From the other hand, the eccentricity can easily be calculated evaluating the polar equation of a conic section at the pericentre:

$$r_p = \frac{a(1-e^2)}{1+ecos0} = \frac{a(1+e)(1-e)}{(1+e)} = a(1-e) \tag{7-6}$$

Finally:

$$e = 1 - \frac{r_p}{a} \tag{7-7}$$

The *pericentre radius* is the module of the position vector relative to the planet that the S/C has at the point of minimum distance (known from the ephemeris data):

$$r_p \approx \left|\vec{r}_{Bepi_{SSB_{eph}}}(t_p)\right| \tag{7-8}$$

If the general Equation (7-6) is considered for a general position r and particularised for the point where the S/C crosses the SOI of the major body (*incoming position modulus*) it is possible to compute its true anomaly:

$$r_{inc} = r_{Bepi\_SSB\_eph}(t_{SOI}) \approx R_{SOI\_major} \tag{7-9}$$

$$\theta = \cos^{-1}\left(\frac{\frac{a(1-e^2)}{R_{SOI_{major}}} - 1}{e}\right) \tag{7-10}$$

This is an approximation, since the crossing obviously does not happen exactly at the right value (the ephemeris data presents a time span of 6 h). Using the true anomaly, it is possible to determine the hyperbolic anomaly and finally the pursued time-of-flight (or period of the orbit):

$$F = 2\tanh^{-1}\left(\sqrt{\frac{e-1}{e+1}}\right)\tan\frac{\theta}{2} \tag{7-11}$$

$$T = \sqrt{\frac{-a^3}{\mu}}(e\sin F - F) \tag{7-12}$$

For the ***ode45*** (MatLab):

$tspan_{bw}$ = T/2:-$step_{size}$:0

$tspan_{fw}$ = 0:$step_{size}$:T/2

Figure 7-12 schematizes the manoeuvre intuitively:

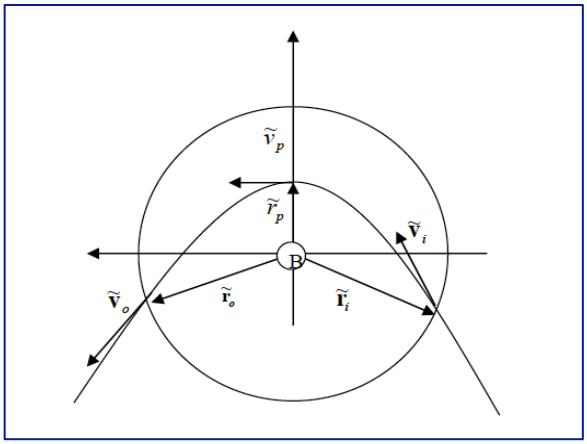

Fig. 7.12 Flyby manoeuvre about the pericentre of a major body [2].

Note: - The tilde symbolizes that the vectors associated to the represented magnitudes should be expressed relative to the swing-by major planet.

Summary of flyby modelling:[110] - To calculate the hyperbolic orbit period (T) the necessary incoming and pericentre states are taken from the ephemeris. On the other hand, the outgoing state is generated by the forward propagation of the flyby itself (last state vector).

Some brief comments prior to execution of the algorithm are: (see **Figure 7-13**).

- To help Cowell's method, the presence of the Earth is neglected during the first 60 integration steps (15 days) to avoid problems in the departure since, although the first ephemeris position is about $3.4 \cdot 10^5$ km from Earth, it is still within its SOI.
- Discontinuities are expected at the link points between the last state obtained by the backward propagation of the flyby and the last state obtained by the heliocentric propagation with Cowell before entering within the SOI of a planet[111]. This is because the applied flyby numerical model – *which despite being a precise approximation is only that ultimately* – uses the ephemeris pericentric state vector. This is especially remarkable for Earth's flyby, since it is the one that needs more time to occur (535 days) therefore accumulating larger errors than others.

[110] Ideally, it would have been desirable to use the last propagated heliocentric state (when crossing the Earth's SOI) as an input to compute the state at the pericentre to define the hyperbola for the flyby, but the inaccuracies introduced by both the perturbation model (Cowell) and the integrator scheme itself did not make this possible.

[111] The last state vector propagated backwards in the flyby model will be very similar to that of the ephemeris that crosses the planet SOI (since they correspond to the same point and the numerical flyby propagation does not have time to produce large errors because it is too short).

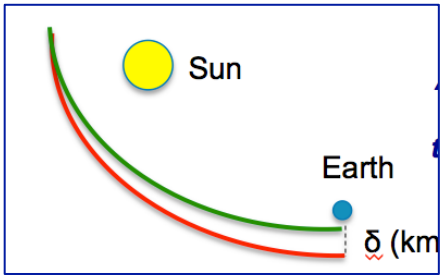

Fig. 7.13 A first point of the ephemeris trajectory (green) crosses the Earth's SOI at a given epoch. At the same epoch, the last point propagated by the Cowell's method (red) is still far away, due to inaccuracies and various types of accumulated errors.

Figs. 7-14 and 7-15 show the attainment of each flyby and both interplanetary trajectories of ephemeris and numerically propagated for the entire simulation.

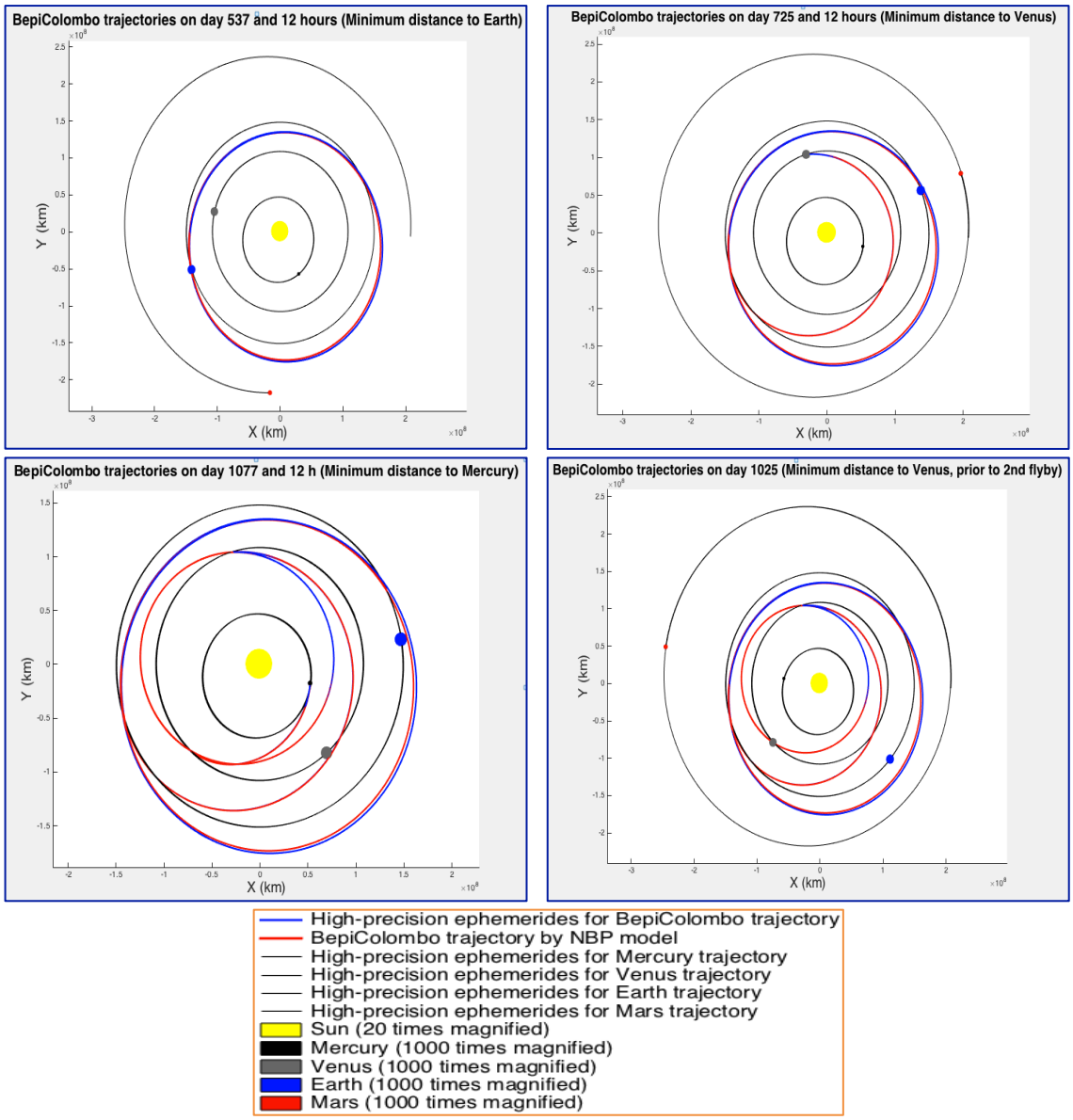

Fig. 7.14 Interplanetary trajectories at each flyby. Clockwise, the S/C at pericentre of: Top left) Earth. Top and bottom right) Venus (two flybys). Bottom left) Mercury.

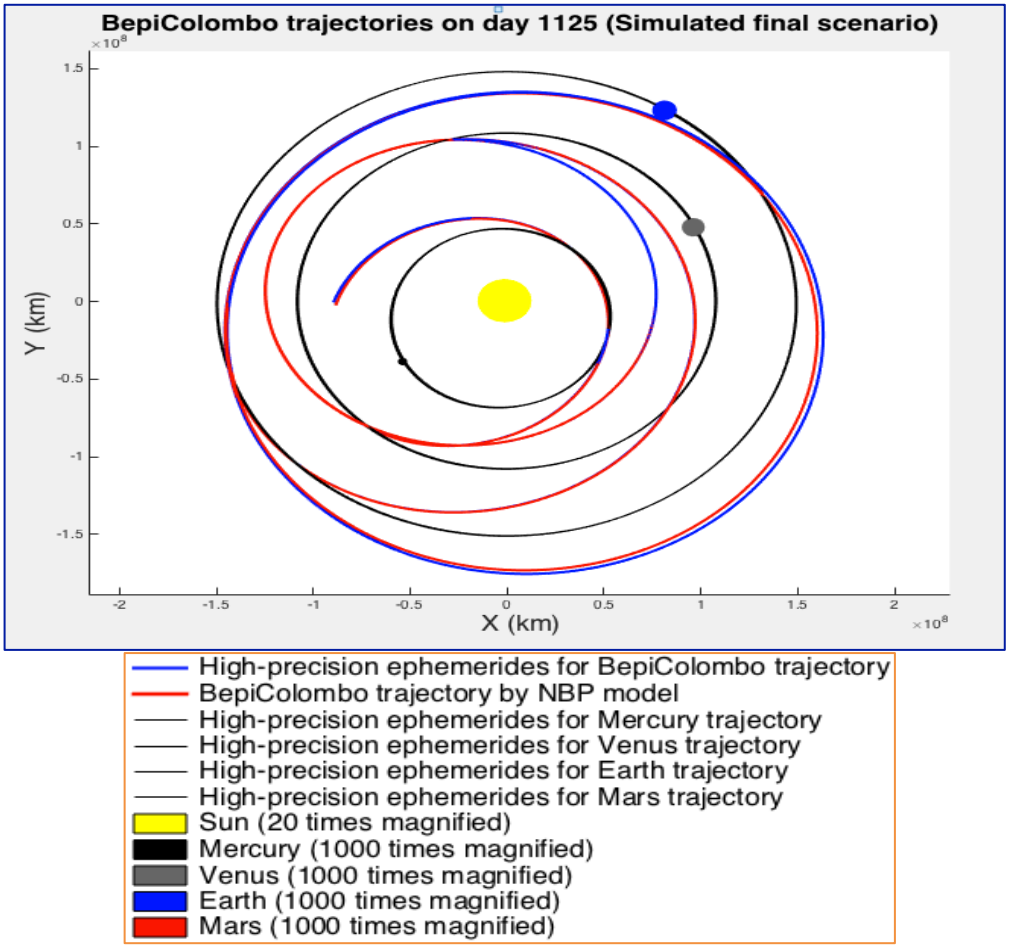

Fig. 7.15 BepiColombo' s interplanetary trajectory computed using Cowell's method. Comparison with the ephemeris. (Simulation time 27000 h = 1125 days).

The sections where one or the other colour is apparently not distinguishable are a consequence of the planar view. By rotating or amplifying the 3D figure that MatLab throws can easily confirm the presence of both trajectories. An acceptable degree of agreement is observed, especially bearing in mind that the perturbation model used is a Cowell's brute–force[112]. Below are the dynamic state vectors obtained at the end of the simulation and those of the ephemeris, respectively. The error is of the order of 1•10^6 km.

$$[x, y, z]_{NBP} = [-8.845876214637183\ , -0.245515728323952\ , 0.819559078673326] \cdot 10^7 km$$

$$[x, y, z]_{Eph} = [-8.958025637214137\ , -0.054130071920104\ , 0.845491761230480] \cdot 10^7 km$$

$$[\dot{x}, \dot{y}, \dot{z}]_{NBP} = [-10.475438113055912, -33.886996669928074\ , -1.335003845628775]\ kms^{-1}$$

$$[\dot{x}, \dot{y}, \dot{z}]_{Eph} = [-11.526508823397720, -33.361467123690012\ , -1.173518839812225]\ km\ s^{-1}$$

[112] It is important to note, once again, that the objective of the validation is to study how the perturbations affect the BepiColombo's interplanetary trajectory, by comparing the results obtained with the full NBP and the partial NBP versions of the equations of motion (see next section). In this sense, the application here of the full NBP aims to obtain a trajectory sufficiently representative of that of BepiColombo's actual ephemeris to guarantee that the trajectory simulated later using the partial NBP approximately crosses the same areas of space. (i.e, implementing only those relevant perturbations indicated by the application of the 2D map to the actual ephemeris trajectory – which was made in the *Section 7.2* in the qualitative study)

On the other hand, as discussed in the results of the 2BP model implemented previously, the trajectory plot itself is not the best way to elucidate the degree of discrepancy with that of the ephemeris. Therefore, **Figure 7-16** shows a graph with the Error Function, which is simply the difference between the trajectories (in terms of distance).

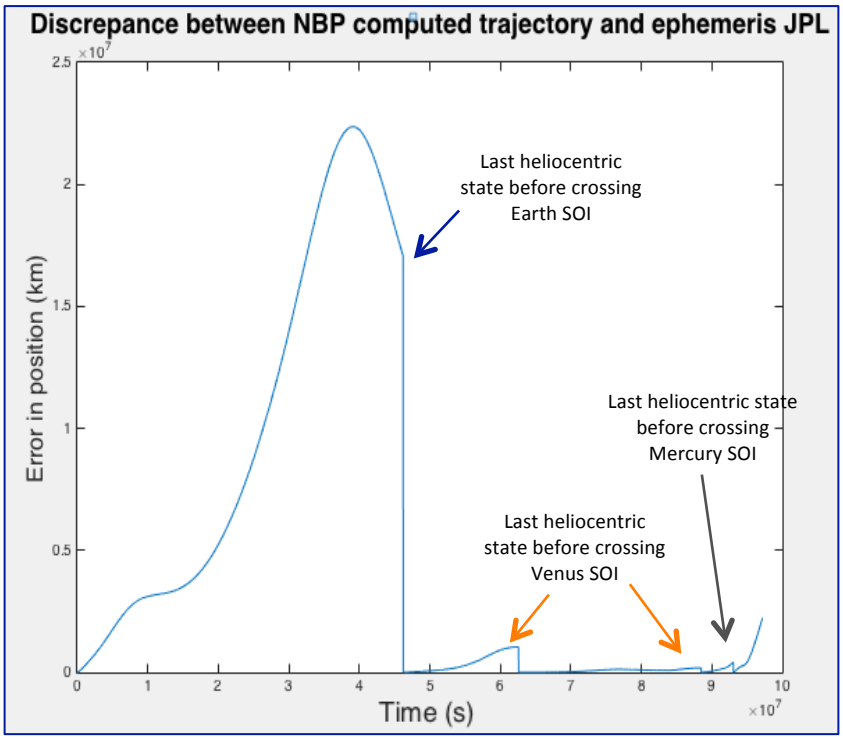

Fig. 7.16 Discrepancy between the full NBP numerically computed trajectory and the ephemeris (Function Error).

Four discontinuities are detected, one for each SOI entry. Their magnitudes are approximately proportional to the time spent in each heliocentric propagation before getting a first state that crosses a SOI in question (they are not perfectly proportional because there are also perturbations acting). As explained earlier, these discontinuities are a consequence of having built the flyby model based on the ephemeris data for both the first state that crosses a planet SOI (incoming state) and the state associated with the minimum distance (use as the pericentre) and because of the propagation drift. After the discontinuity of the first flyby, the discrepancy is reduced in almost two orders (from an average of $1 \cdot 10^7$ km to an average of $2.42 \cdot 10^5$ km).

The question that may arise is *why that large error in the first part before reaching Earth's SOI*? Numerical integrators introduce errors "*per se*" that are difficult to handle, since many of the desirable qualities, when choosing the most suitable one, are opposed to each other and cannot be achieved to the fullest extent (e.g, large and variable step–sizes, stability to exponential error growth or insensitivity to round off error). The main sources of error in numerical schemes are shortly described below:

Round off errors: - A computer can carry only a finite number of digits of any number, so at each operation one or more of the last significant digits are rounded. Repeated many times in the integration process can lead to considerable errors. [B.4]

According to Brouwer and Clemence [43] the probable error after n steps – in number of decimal places – is given by log [.1129$n^{3/2}$]. The last state before crossing Earth's SOI (step 2140) would give a value of 4.048. Since HORIZONS provides its ephemeris states with 16 significant digits – which coincides with the maximum given by MatLab in double precision arithmetic – at least 16 + 4.048 ≈ 20 places should be taken in the calculations in order not to lose the information in each operation (which is not possible when working in double precission with MatLab). So, the lesson here is that the fewer integration steps, the smaller accumulated round off error.

Truncation errors: - It is a result of an inexact solution of the differential equation due to not using all of the series expression employed in the integration method. The larger the step – size, the larger such error. Thus, here the ideal is to have **small step – sizes.**

Since both type of errors are antagonistic and inherent to any numerical scheme, they are unavoidable and the aim is to use a method that minimises their sum. Moreover, as discussed in Chapter 3, Cowell has the advantage of simplicity of formulation and implementation and allows any number of perturbations to be handled at the same time but, in systems with a large dominant central body (such as the Sun) many significant digits must be taken due to the large difference in forces between that central and the perturbing bodies. On the other hand, when motion is near a large attracting body, small integration steps must be taken that result in an aggravation of the round off errors (even working with double precision).

Hence, *how do we finally interpret that large and growing error* shown in the first part of **Figure 7-16**? Firstly, the morphology of the curve (error that increases exponentially over time) responds quite well to the characteristics of a round off type error. It cannot be fully guaranteed, but in principle it should dominate over truncation, as the step size of 6 hours seems quite small[113] (out of a total of 27000 hours considered for the whole simulation).

7.4.2. Improvements provided by Encke's method

Encke's method was already conveniently explained in **Chapter 3**. What is going to be done next is to compare the results offered by the two methods (Encke and Cowell) up to the first flyby, as this is where Cowell's biggest accumulated error is found, due to its exponential tendency to accumulate rounding error because it is the longest simulation leg.

[113] As commented, the numerical integrator implemented in the main algorithm and MatLab functions was a Runge-Kutta – Fehlberg scheme of orders 4,5. It has a relatively small truncation error and step-size is easily changed. However, one of the disadvantages is that there is no simple way to determine such truncation error, thus being difficult to determine the proper step-size. A consequence of this is that it uses only the last calculated step each time.

Once the initial conditions have propagated, **Figure 7-17** shows the simulated trajectories. It can be seen that Encke's method reproduces the ephemeris trajectory much more accurately, which is especially noticeable in the half-right part (although it is not the best plot for it). By its part, **Figure 7-18** contains both *Error Functions* (discrepancies with respect to the ephemeris trajectory). Both figures are displayed **before crossing Earth's SOI** (535 days).

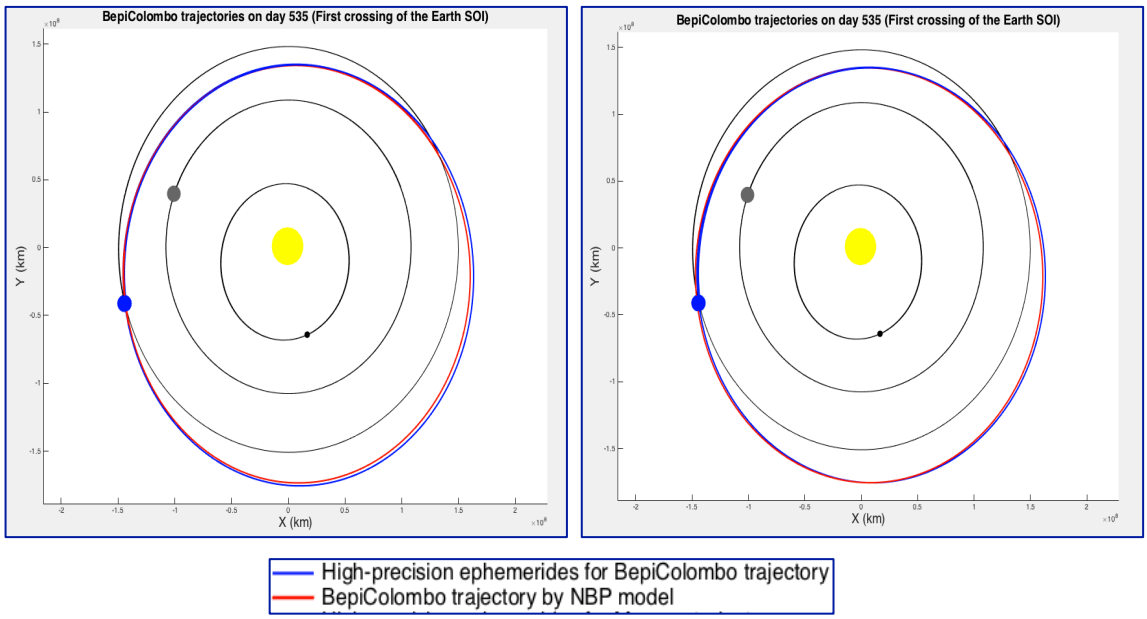

Fig. 7.17 Comparison between full NBP propagated trajectories and ephemeris using: Left) Cowell. Right) Encke.

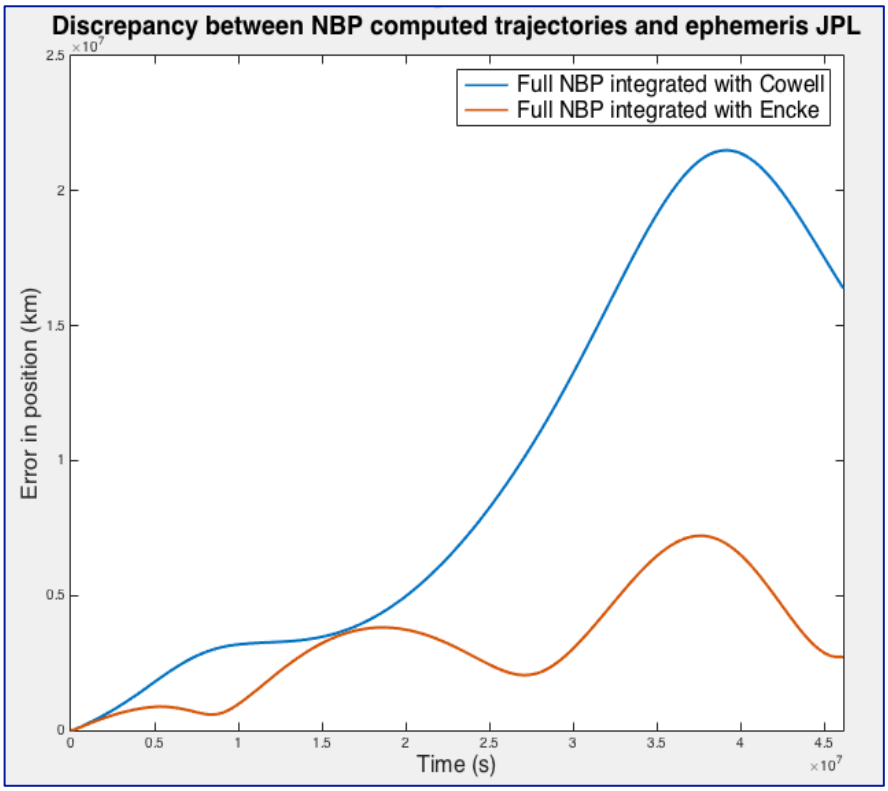

Fig. 7.18 Comparison of discrepancies with ephemeris trajectory until first crossing of Earth's SOI. Encke's method is implemented with an integration step of 24 h.

What is observed is, that, as a result of taking larger integration step-sizes[114], Encke reduced by an order of magnitude the discrepancy in its maximum, average and final state (when penetrating the Earth SOI) with respect to Cowell. This advantage, allows better detection of oscillations that occur when the simulated trajectory approaches and moves away from the real ephemeris trajectory (especially in the first two valleys of the beginning, when the order of magnitude of the discrepancy is still small). Nevertheless, when Cowell is used, due to the large accumulated numerical error, this is not possible until the final (and closest) approach to Earth takes place.

Finally, a question arises: *continue with Cowell or switch to Encke*? As it seems trivial, the choice has no colour, but there are drawbacks related to using Encke for this particular case (because of the peculiar trajectory). Are the next:

- Encke leads to a reduced discrepancy. However, due to the morphology of this ephemeris trajectory, continuing to use this method would require multiple rectifications, which would include the creation of new osculating orbits every time a SOI was crossed (both in and out) forcing also to change the dominant body in the functions used to build it (see **Chapter 3** subsections 3.3.2 and 3.3.3) which entails a greater computational cost

- To carry out the entire propagation autonomously, it would require a high accuracy in the detection of the first state that crosses the Earth SOI to ensure that the S/C is at the correct distance at the right epoch, which means that using 24 h as an integration step-size is not adequate (which was its advantage). Alternatively, if we choose to use the numerical model to construct the hyperbola – *as in Cowell's case* – it would be necessary to reuse the ephemeris states to build a suitable pericentre (then, it would be the same case, with a smaller discontinuity but still large, thus not justifying its use either)

- The rectification condition within Encke's algorithm was not used throughout the execution for the chosen integration step size, which means that it might still be possible to choose a larger one to further reduce the error. However, that is not desirable, as it would imply the need to obtain the planetary ephemeris with the same time span, which would lead to a lower detection sensibility of SOI crosses (same problem as before)

Following this rationale, Cowell is preferred because it is simpler and in the end what matters is to compare what happens when bodies are removed from the perturbing term (thus, the same numerical error is expected in any case of NBP analysed, and the observed differences would only be the result of considering more or less perturbations).

[114] In **Figure 7-18**, the blue curve (Cowell) has been obtained with an integration step-size of 6 h (see subsection 7.4.1) while the orange (Encke) has been obtained using 24 h. The reason why they could be represented together is because the data matrix obtained for Cowell (i.e., containing the discrepancies) has logically four times as many rows so they have simply been represented the values every 24 h to match with the Encke ones.

Additional observations

In addition to everything discussed or analysed before, it is important to keep in mind that what has been reproduced is an ephemeris trajectory developed by a NASA platform, which is like a "*black box*" from the perspective of a standard user. Following this reasoning, it is possible that other forces or events other than gravitational ones have been considered (to which the user cannot have acces). Among the most relevant, it is worth mentioning:

- **Possible correction manoeuvres (ΔV) planned for specific points within the trajectory** ➔ Difficult or impossible to know for a standard user (see **Chapter 2** *subsection 2.3.1*)
- **Solar radiation pressure** ➔ It would require further deepening to get an idea of the order of magnitude it could exhibit, depending on the size of the spacecraft (it is not very likely for a massive spacecraft like BepiColombo)
- **Asymmetry of the celestial bodies** ➔ The concept of non-spherical Earth (also for other planets) is commonly applied to planetocentric orbits. In this case, the S/C passes very close to the Earth, Venus and Mercury, but it does so very fast (only for a few hours[115]). In any case, it might be implemented as well.
- **Other instantaneous external or internal** forces present at some points

It is worth mentioning that not considering perturbation effects such as those described above would not really affect the comparison of the results obtained when propagating with or without all the bodies, since in both cases (full or partial NBP) it would be carried out under exactly the same conditions (and it has proven that BepiColombo's interplanetary ephemeris trajectory has been reproduced by full NBP with a sufficient representativeness as to accept that the S/C crosses the same areas in space than the actual ephemeris does).

Note: - Such alternative sources of perturbations have been presented in order of probable influence. Others as atmospheric drag are not relevant (since the first ephemeris data is $3 \cdot 10^5$ km from the Earth and its maximum point of approach to the Earth for the flyby is of the order of 10^4 km; in the case of Mercury, it has an extremely thin atmosphere). Magnetic fields were as well discarded directly.

[115] BepiColombo is scheduled to pass about 11000 km from Earth for its flyby on 13th April 2020. However, taking into account the time it will be at that mínimum distance (24 h máximum, since a specific day is quoted) that disturbance effect is not expected to be significant.

7.5. Trajectory propagation using a partial NBP model

To validate the 2D maps that show the perturbed gravitational acceleration field – developed in **Chapter 6** – it is necessary to compare the results obtained for both full and partial NBP trajectory propagations. Some important points to remember are:

1) In both cases, propagation shall be performed using the same perturbative method (Cowell). Thus, the same degree of numerical error is expected in them (allowing comparisons).

2) The so-called "partial" NBP is the same propagation as the one carried out in the previous section, but it includes in the perturbing term only those planets in which the S/C crossed[116] their respective perturbative spheres (therefore, the same algorithm is used but this time executed neglecting the presence of bodies not marked as relevant for BepiColombo's ephemeris trajectory).

3) The appellation "static" refers to the fact that the inclusion of those relevant planets is taken for the entire simulation (despite the S/C spends only a specific time within each PS).

Considerations prior to code execution

- To better discriminate the details, the Error Function diagram will be divided into two parts: one to the point before the Earth SOI crossing and the other from the next point (i.e, once within that SOI) to the end of simulation. The objective is to avoid the first discontinuity

- A mathematical "trick" is done to include in the comparison a modified version of the 2BP. Given that the three gravity assist manoeuvres that appear in the simulated trajectory barely involve 36/4500 integration steps (9 days over 1125 in the simulated time) it can be assumed that these events are negligible in terms of time (being the magnitude that most influences the deviation capacity of a perturbative acceleration). Hence, it will "artificially" include the needed flybys, but not their responsible planets for the rest of the time (thus allowing to test what happens when perturbations are not taken into account but the S/C is in the "correct" course).

The three Error Functions (i.e., discrepancies with the ephemeris trajectory) are displayed together for the whole simulation time (1125 days) in **Figure 7-19**. The first thing that stands out is that the curve associated with the modified 2BP exhibits lower exponential growth in the first section (also a smaller maximum). Given that qualitatively all curves have a similar behaviour in time it suggests that such a difference is probably due to the numerical error of rounding (accentuated by Cowell, as explained earlier).

[116] They are *Mercury*, *Venus* and the *Earth* (see Section 7.2)

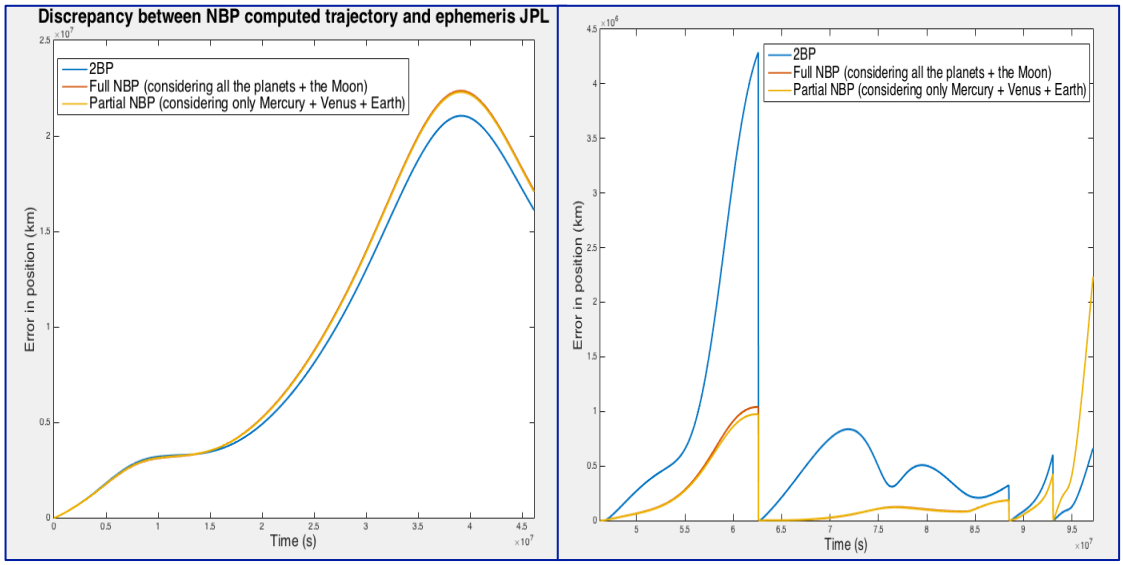

Fig. 7.19 Comparison of discrepancies with ephemeris trajectory for the entire simulation using full NBP and partial NBP with the selected celestial bodies. Notice that the Y-scale changes in both pictures as it corresponds to different sections (i.e, before entering the Earth's SOI and from there to the end of the simulation). It includes a modified version of 2BP.

It is possible that, during the 535 days until the first ephemeris state crosses the Earth's SOI, the perturbations produced by the planets are not really significant, otherwise some irregularities in the NBP curves should be observed with respect to the 2BP. Anyway, it is only a hypothesis and with the high numerical error existing, this is not a good stretch to draw conclusions. As regards the total and partial trajectories of NBP, no differences are observed, which also supports the fact that introducing perturbations in the propagation with Cowell is the reason for this greater discrepancy.

Deepening now the right-hand graph of the figure, the curve associated with 2BP is practically always placed above the other two. Its mean value of discrepancy with the ephemeris in this section is $6.69 \cdot 10^5$ km, while the average for NBPs is about $2.37 \cdot 10^5$ km, being almost three times lower (we must bear in mind that we are talking about deviations produced by the consideration or not of disturbances, always lower than those due to numerical errors). Each time the S/C moves away from an SOI (once a flyby ends), the 2BP experiences a greater growth in its discrepancy, except in the case of Mercury. This last case could be due to the fact that, after the second flight of Venus, the S/C leaves behind (and further away each time) those planets, which were theoretically the most disturbing[117]. The conclusion regarding the comparison between 2BP and both versions of NBP is that *considering perturbations is crucial to adequately reproduce the real ephemeris trajectory*[118] (as expected on the other hand).

[117] The larger numerical error related to Cowell's propagation is still present and once the S/C is near Mercury, its trajectory is clearly governed by the Sun. In consequence, in that moment the S/C is only significantly subjected to the perturbation of Mercury itself (thus not very affected by perturbations but rather by the numerical error).

[118] It should be noted that the scale of the discrepancies within this section (i.e, from the Earth's SOI penetration to the end of the simulation) are more than an order of magnitude lower, which indicates that here it is possible to carry out the analysis of perturbations.

In order to confirm such analyses and the subsequent conclusion, it might be interesting to check how the each component of the state vectors (positions and velocities) change over the simulation time. **Figure 7-20** shows it for the second section, since there is where the significant differences have been identified and also where it is valid to carry out the analysis.

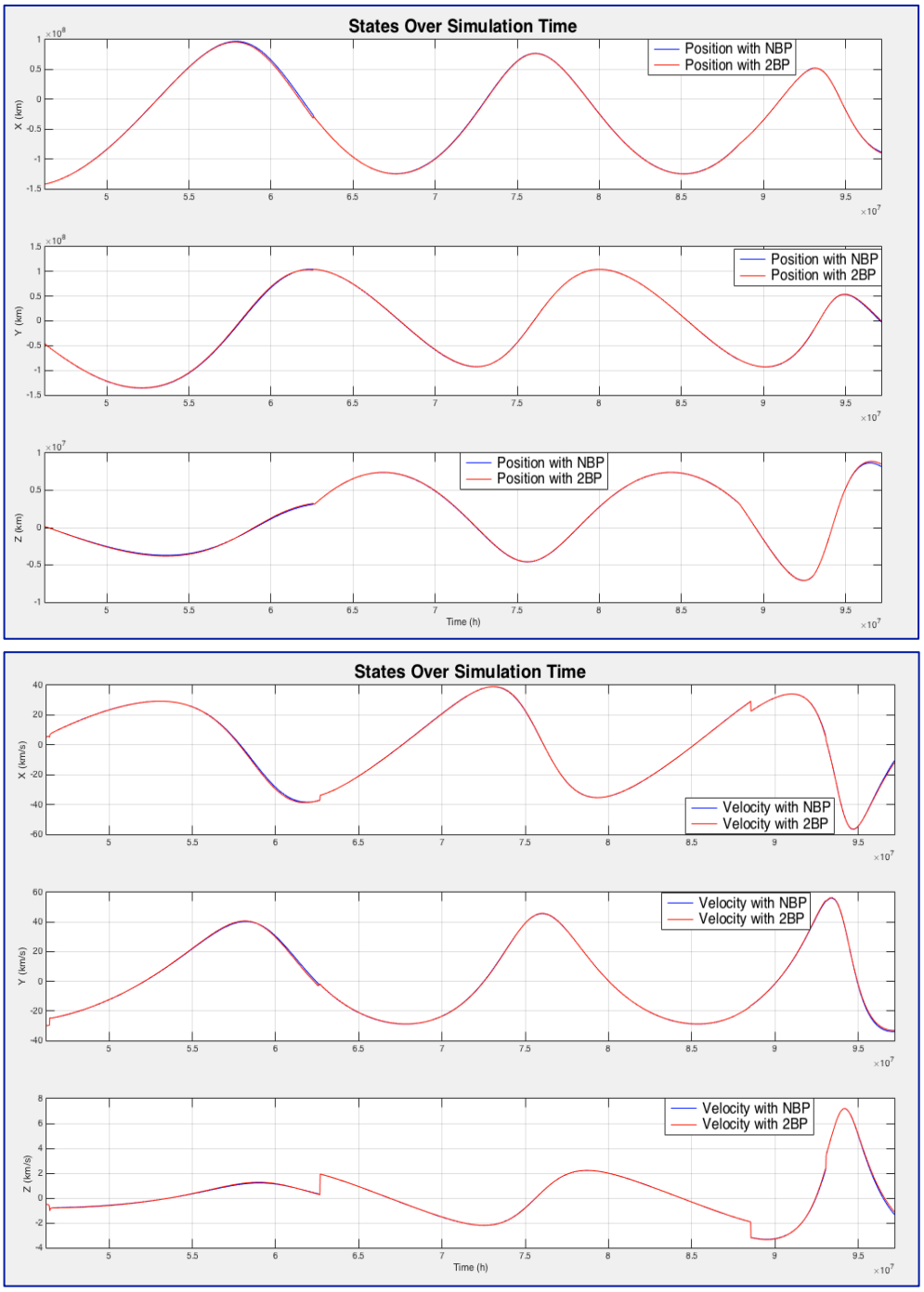

Fig. 7.20 Comparison of state vectors over time (individual components). 2BP – NBP.

It can be observed significant differences in both types of state (positions and velocities) as shown by the clear staining of the blue colour over the red. Taking into account that gravitational perturbation are small in nature and need to act for a long times to produce large deviations, their influence on interplanetary trajectories is then demonstrated.

Focussing now in the results associated to each NBP propagation, it can be observed that, both curves coincide highly except for a slight difference in a fragment between (7.5-8.5)•10^7 km (where orange is distinguished from red) and just before the first flyby on Venus, where the difference is only 6.71•10^4 km (it is important to note that the scale at Y axis has been reduced due to the division of the diagram). The greater discrepancy in the full NBP is not significant (due to the value) and may be due to Cowell handling more number of bodies or, possibly the inclusion of the Moon (which has not been considered in partial NBP). Regarding the change in states vector over the simulation time:

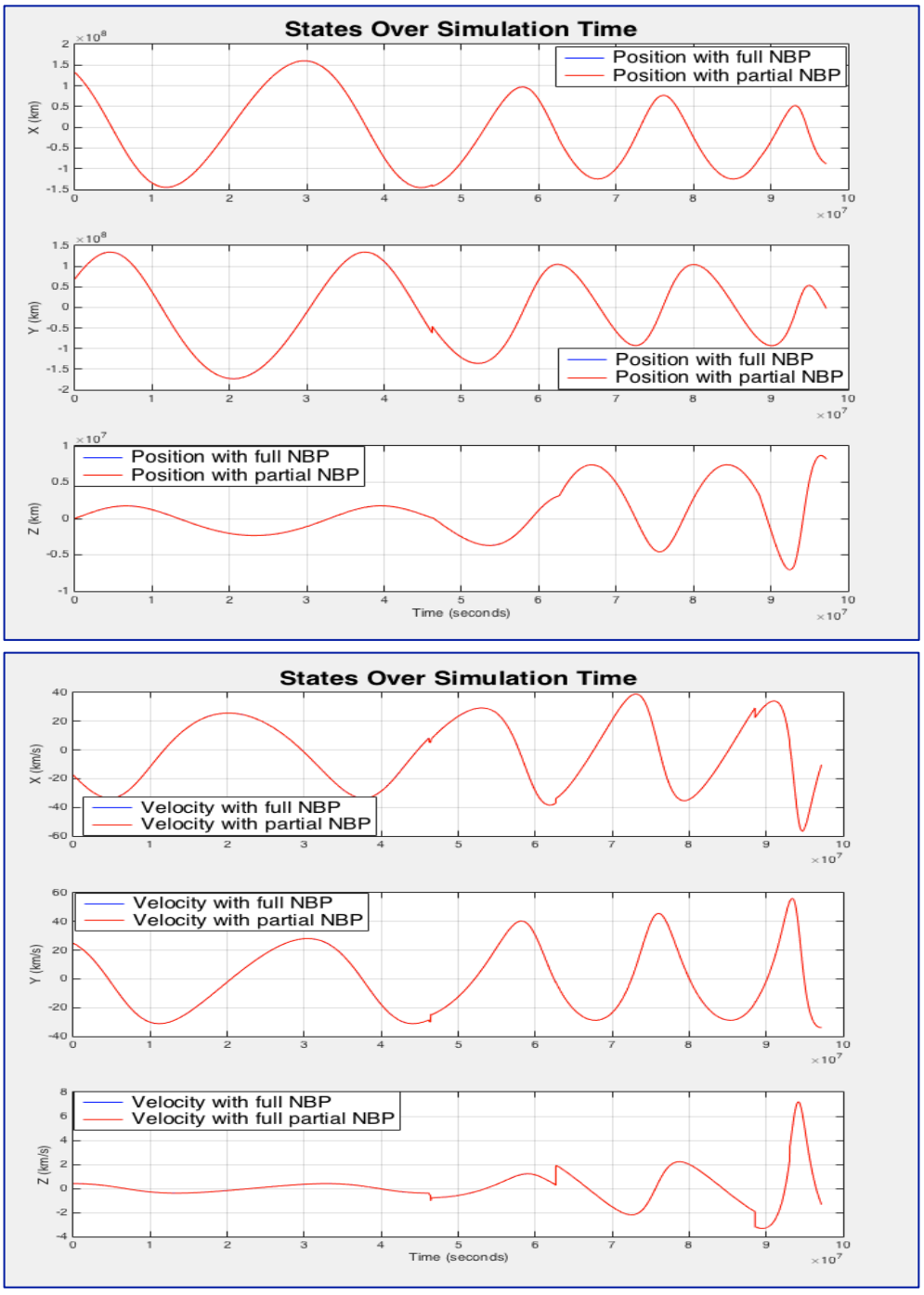

Fig. 7.21 Comparison of state vectors over time (individual components). NBP – NBP.

No significant differences are observed (it seems that only one curve appears, which happens because MatLab cannot plot two curves with such a degree of coincidence, but it appears at some point if enough zoom is made). The conclusion is that, *removing those planets where there is no crossing in their perturbation spheres, does not affect significantly to the accuracy of the trajectory*.

A last representation is to be made to test what happens with the difference of discrepancies with respect to the ephemeris trajectory, i.e., by computing the difference between full and partial NBP curves of **Figure 7-19**. **Figure 7-22** shows it.

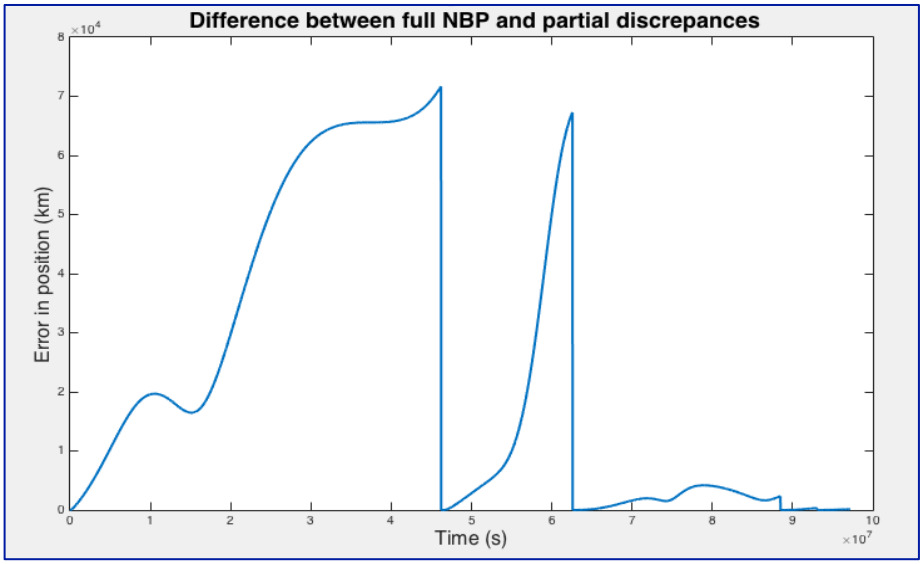

Fig. 7.22 Difference between full and partial NBP discrepancies.

What can be seen first of all is that the shape is very similar to that presented by each of the curves associated with the discrepancy with respect to the trajectory of the ephemeris. This fact, together with the fact that the strong growth of the variation between curves takes place when the spacecraft approaches a planet, demonstrates that these differences can be due fundamentally to the numerical error when propagating because, although both use Cowell, it suffers more rounding error the more bodies it handles. On the other hand, the biggest difference in the discrepancies corresponds to about 70000 km, which many correction manoeuvres carried out in the middle of the mission far exceed. Therefore, it can be accepted that *there is no significant difference between reproducing the BepiColombo ephemeris trajectory considering all the planets of the solar system as perturbations or only those that the 2D maps indicate as relevant*.

With these results in hand, the information provided by the 2D maps – *regarding the relevance of bodies whose perturbations influence the trajectory of any spacecraft with propulsion capabilities similar to BepiColombo* – can finally be **considered valid**.

7.6. Trajectory propagation using a dynamic partial NBP model

At this point, the validation of the 2D maps has already been carried out. Therefore, an additional validation is no longer required. Beyond that, of course, a homologous development but using the information from another known mission whose S/C exhibits a propulsion plant similar to that of BepiColombo, it would always be a good idea to double check).

However, the form of propagation followed so far is "*static*" and has a disadvantage: its application required a reference trajectory. I.e., to include the relevant selected planets in the disturbing term of the NBP equation of motion, it was necessary to examine the ephemeris trajectory first rigorously in order to properly detect which Disturbance Spheres were crossed and which were not (which has been done in *Section 7.2*).

A more general and automated application of the information provided by the 2D maps could be destined for ***efficient autonomous navigation***, which means that these relevant planets would be hypothetically included in the disturbing term, but only during the integration steps when the S/C is within their respective PS (in other words, only at the moments and places where the disturbance threshold is exceeded). To verify whether such efficient propagation of trajectory leads to successful results (i.e, to reproduce BepiColombo's ephemeris trajectory without losing the accuracy demonstrated by the static integrator), a dynamic numerical integrator will be created and tested.

How such "dynamic propagator algorithm" works? It is sensitive to the instantaneous position that the body occupies at each point of its simulated trajectory. It is simply based on the numerical integration of the differential equations of motion, but also computes the vector sum of all perturbing accelerations in each integration step. When it detects that the established threshold is exceeded – in this case the one fixed according to the propulsive characteristics of a BepiColombo type S/C – it discriminates the PS that has been penetrated and includes the responsible bodies in the perturbative term for the next integration step. Its main advantages are:

+ Stores time and distance travelled in sections where the threshold is exceeded in order to carry out a post-propagation sorting of the bodies responsible for the deviation with respect to 2BP (this is due to the time of action of the perturbative acceleration and, if several bodies were included for the whole propagation, this sorting could not be carried out).

+ It allows saving computational cost, not making necessary a propagation considering perturbative terms for all integration steps.

Figure 7-23 shows an implementation scheme containing the main functionalities of the algorithm with further detailed description.

$$\ddot{\mathbf{r}}_{12} = -\frac{G(m_1 + m_2)}{r_{12}^3}(\mathbf{r}_{12}) - \sum_{j=3}^{n} Gm_j\left(\frac{\mathbf{r}_{j2}}{r_{j2}^3} - \frac{\mathbf{r}_{j1}}{r_{j1}^3}\right)$$

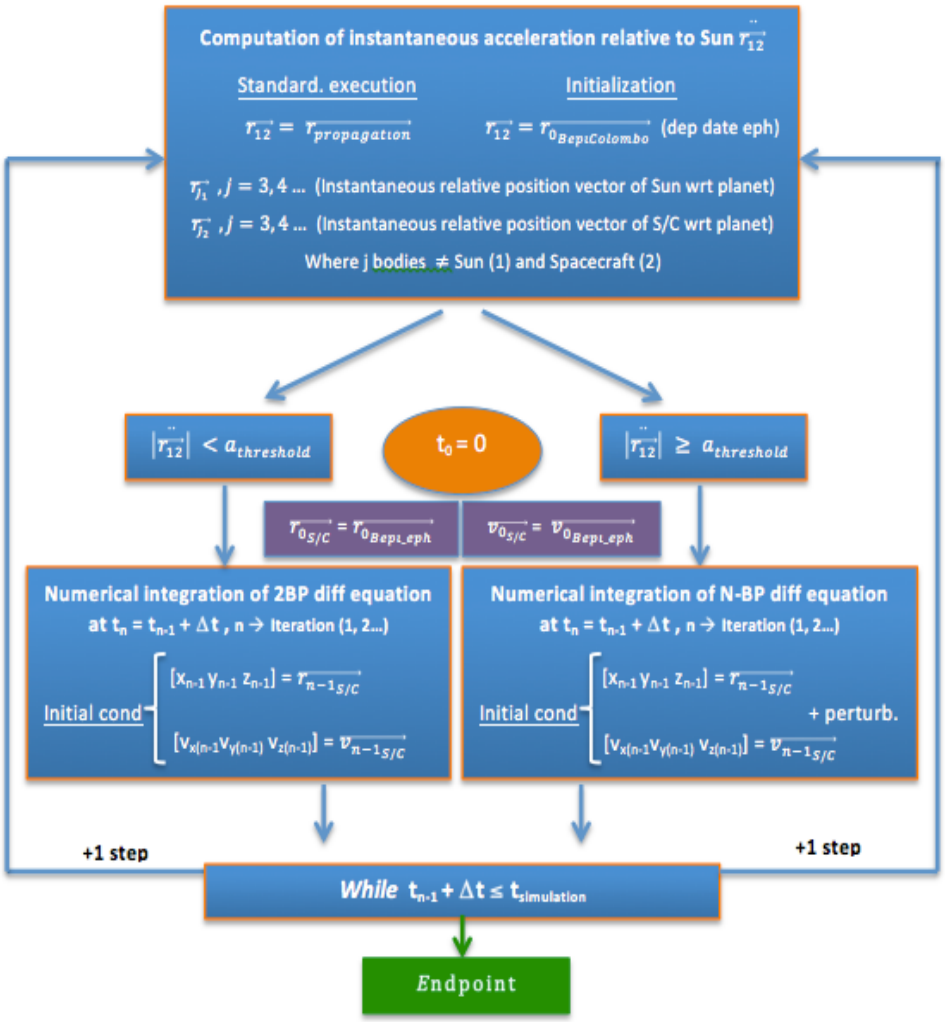

Fig. 7.23 Operation of the dynamic propagator algorithm.

Algorithm operation

1) The main code calls the planetary ephemeris (and those of BepiColombo) and defines the necessary ancillary data.

2) Before the first integration step takes place, the code should check if the S/C is at initial conditions within a PS. However, as mentioned earlier, the Earth is neglected during the first 15 days (360 steps) to avoid problems with Cowell at the departure. Thus, starts with 2BP.

3) It continues with the basic integration 2BP and checking the magnitude of the perturbing term for each new step, until the threshold is exceeded. At that point, it computes the distances to each body, detects the PS that has been penetrated, stores its associated body and shifts to partial NBP propagation, including it in the perturbing term.

4) It maintains that type of propagation during the necessary steps, shifting to 2BP when appropriate.

Before jumping to the obtained results, some preliminary considerations must be stated. **Figure** 7-24 contains such results.

- 2D maps were created based on a grid that was considered only just a little beyond the orbit of Jupiter, since it corresponds to the desired part as the object of study of the solar system (hence, up to a certain distance from the Sun where later developing trajectory propagation applications to show the strengths of using such maps)
- Based on that, the code includes the PSs of all these planets to allow double-checking in real time about, if in reality, the S/C never penetrates other than those of Mercury, Venus and Earth (as demonstrated before with the fact of not finding significant differences between the discrepancies using total or partial NBP). Therefore, it could act as a second "*double check*" indicating that the information provided by the 2D maps is correct and reliable

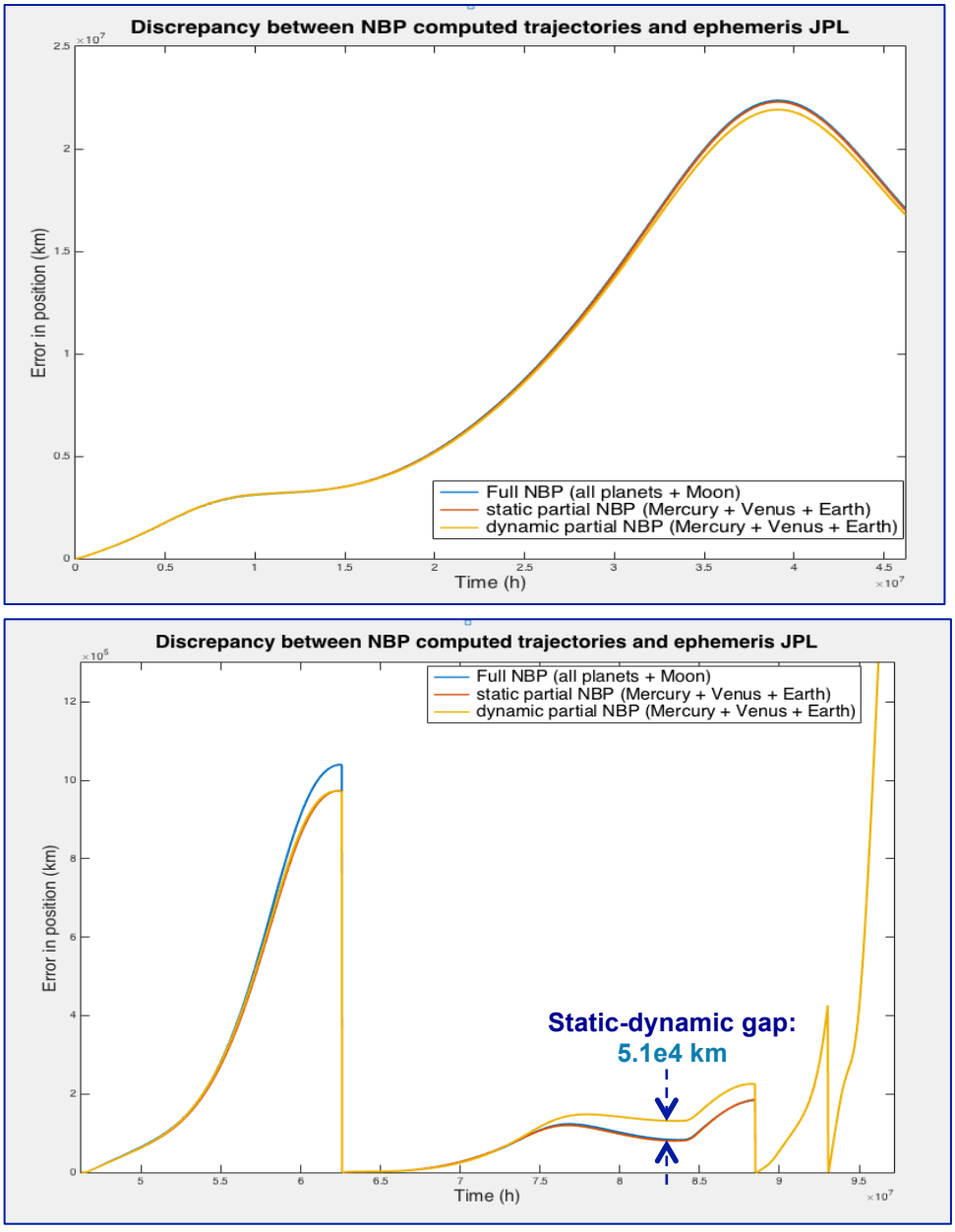

Fig. 7.24 Operation of the dynamic propagator algorithm.

It is observed that the dynamic solution exhibits a slightly smaller discrepancy in the first section with respect to the ephemeris trajectory than the static one (picture of the top). It is due to the algorithm using 2BP during the integration steps where the S/C is does not undergo a perturbing acceleration exceeding its threshold. Thus, it leads to a smaller round off error, since the static solution includes the relevant planets in the perturbing term all the time during the propagation).

In the second stretch, the static and dynamic NBP match very well, except for a fragment in which they begin to differ, corresponding to when the S/C is outside the Venus PS before entering again. It indicates that during this phase, the change to 2BP should not have occurred, suggesting that Venus PS is actually larger than what was calculated from the 2D maps. Nevertheless, it should be mentioned once again that this interpretation implicitly assumes that the S/C would apply a thrust to cancel the total perturbation when the threshold is not exceeded, which would allow to admit that it does not exist (which obviously is not true). But said thrust was not considered in the equations. **Figure 7-25** shows the change in state vectors over the simulation time.

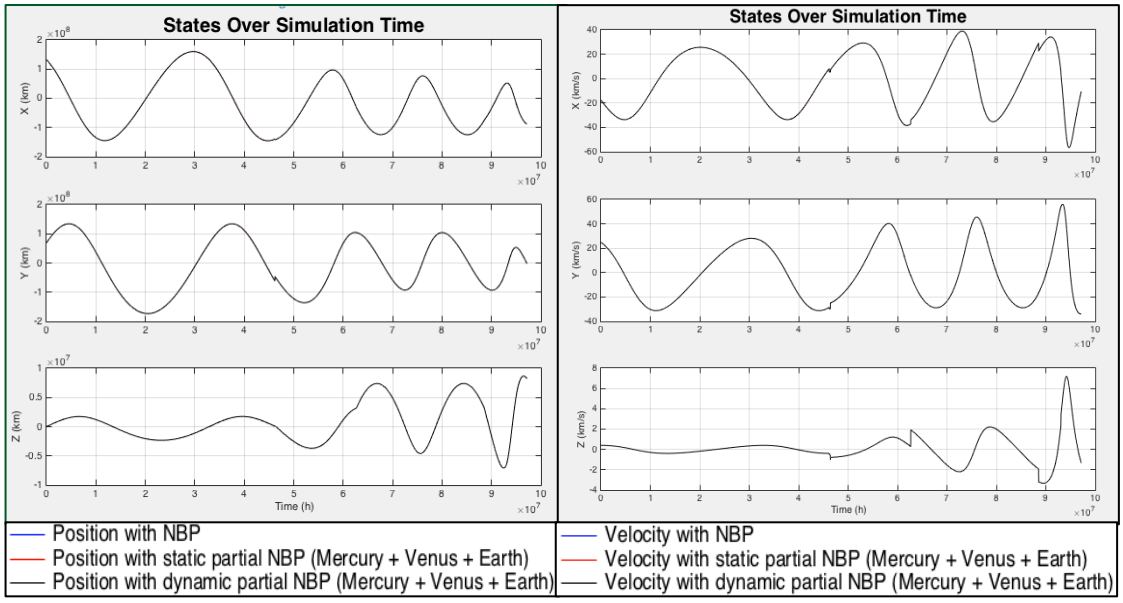

Fig. 7.25 Comparison of state vectors over time (individual components). Full-partial.

Again, as happened when comparing partial NBP with full NBP, no differences are observed (apparently there is only the last curve, which happens because MatLab cannot plot three curves with such a degree of coincidence, but it appears at some point if enough zoom is made) The conclusion is that, *the dynamic propagator is capable of efficiently reproducing the ephemeris trajectory and can be considered as valid.* **Figure 7-26** condenses the information leading to its strengths and advantages. It shows the time spent by the S/C within each Perturbation Sphere. In order to differentiate to which planet they correspond, a coded number has been provided to each of them (ranging from 1 to 7 and reserving 13 to identify if the S/C entered within two PS at the same time).

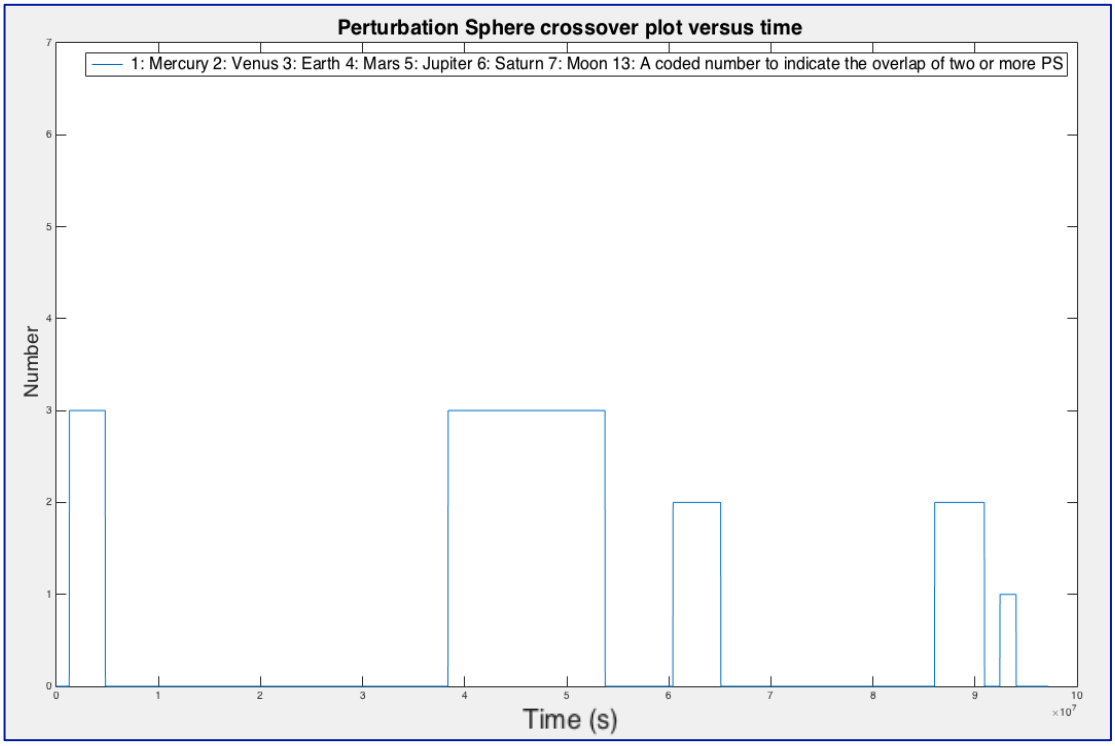

Fig. 7.26 Perturbation Sphere crossover plot versus time.

Note: - Remember that Earth begins to be considered after the first 15 days of the propagation initiation.

The following observations were made:

- The gaps correspond to moments when the integrator applies 2BP
- PSs crossing for planets beyond Earth were not detected
- No overlap of two or more PS occur
- Times for each PS crossing (expressed in seconds here) coincide very well with those obtained for the ephemeris

And finally, summarised, here there is the advantages that its validation offers:

+ Not need of reference trajectory knowledge (ideal for efficient autonomous navigation)

+ Confirms whether the S/C enters (or not) in other PSs than those predicted by 2D maps

+ Reduces the accumulated round off error due to the use of 2BP in many parts where the inclusion of perturbations is not necessary

+ Saves time and computational resources (less data and faster executions to perform)

7.7. Validation outcomes

Before jumping to the final conclusions chapter, a brief result will be shown, which constitutes the most important milestone achieved by the entire process developed to analyse which perturbations can be neglected without loss of accuracy in the calculation of interplanetary trajectories: the reduction of time and resources.

To do so, a simple debugging has been carried out in MatLab to compare the computational execution times of the main code for full NBP and static partial NBP (which has meant the greatest time savings and, therefore, the greatest success achieved with the use of the 2D maps).

Relevant considerations

• A parallel main code is created – *renaming the original* - to perform the cited debug. In it, some lines have been removed and some function calls reconfigured due to the fact that the partial NBP (when selected by the user) does not need all the planetary ephemeris. The called "***Model_2BP_BepiColombo_V5.m***" is the original and contains all versions (2BP, full NBP, static partial NBP and dynamic partial NBP).

• The first line in the debug profile provided by MatLab corresponds to the total time elapsed in the main code, which calls other functions

• The function called "***Planet_trajectory_HORIZONS_BepiColombo.m***" which is used to create the ephemeris data for each planet, is contained in "***asteroids_cases.m***", this being a more general one in charge of processing the required data from HORIZONS (not only for BepiColombo but for the asteroids analysed before)

• The reason why ode45 is called a few times more than the static propagator function is because the flybys were modelled separately by another function that also uses ode45

Figure 7-27 shows the results.

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
Model_2BP_BepiColombo_V5	1	298.792 s	6.465 s	
asteroid_cases	1	215.322 s	0.000 s	
Planet_Trajectory_HORIZONS_BepiColombo	10	179.152 s	69.471 s	
ode45	4496	75.128 s	31.765 s	
static_propagator	4465	72.590 s	0.945 s	

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
Main_Debug	1	170.787 s	3.426 s	
asteroid_cases_Debug	1	88.142 s	0.000 s	
ode45	4496	77.171 s	32.891 s	
static_propagator_Debug	4465	74.810 s	1.007 s	
Planet_Trajectory_HORIZONS_BepiColombo	3	50.263 s	19.767 s	

Fig. 7.27 Comparison of computational times for full NBP and partial NBP.

The total time for executing the main code has been reduced by **42.8 %** using only the relevant celestial bodies and neglecting the rest. The main reason lies in the **59.1 %** reduction in time that the function ***asteroid_cases.m*** consumes. The ***Planet_trajectory_HORIZONS_BepiColombo.m*** function is now called only three times and consumes **71.9 %** less time than before. That is the great achievement of 2D maps, now there are only three planets from which to obtain ephemeris states in real time (necessary to properly apply the perturbing term in the NBP motion equation).

On the other hand, there are two analogous functions that are always needed regardless of the version used. They are “Sun_trajectory…” and “Asteroid_trajectory...” that serve for gathering the information of the ephemeris from the Sun and BepiColombo, respectively in this case. They are not shown in that list.

CONCLUSIONS AND FURTHER DEVELOPMENT

The main goal of this thesis was to deepen the understanding of the influence that the various perturbations exert on the trajectory calculations performed by the JPL ephemeris model to provide high-fidelity ephemeris in support of spacecraft navigation and other activities related to Solar System bodies. One of the drawbacks behind such a complex method are the large machine-oriented technical requirements it has in terms of time and computational cost. The algorithm that integrates the full dynamical model of the N-Body differential equations of motion requires a continuous update of the dynamical states (positions and velocities) of a large number of bodies for each new integration step, regardless of the perturbation model applied or the nature and diversity of the sources of perturbations other than those gravitational considered.

In reference to the cornerstone of the problem to which this work is concerned with, interest has been directed especially to the gravitational implications of interplanetary trajectories, based on the fact that within its main and longest phase (the cruise phase) a spacecraft is essentially affected by a multi-body attraction, making it necessary to handle a large number of bodies as mentioned above.

To overcome the objective, a procedure has been conducted to build a tool that, applied to any given interplanetary trajectory, has the ability of predicting which gravitational perturbations are not relevant and can thus be neglected without a significant loss of accuracy. The specific steps and milestones that were successfully achieved to develop the tool were the followings:

- A concept of perturbing acceleration threshold was introduced to allow the definition of the magnitude limit of the vector sum of all the gravitational perturbing accelerations that a propelled object can admit without significantly affecting its motion. For the sake of achieving a representative of the modern spacecraft missions, it was calculated based on the electric propulsion plant of Bepicolombo.

- A planar control volume containing a spatial discretisation of the part of the Solar System intended for study was generated and mapped to analyse in which areas the representative spacecraft, accelerated with respect to the Sun, would be subject to planetary perturbations above its threshold. A criterion based on the concept of Perturbative Sphere was defined to allow such identification for any interplanetary trajectory performed or planned with that spacecraft.

- Validation tests based on ephemeris of the interplanetary trajectory planned for BepiColombo's mission proved that no significant differences are found when comparing the discrepancies exhibited by a numerical propagation of its initial conditions with the N-Body differential equations of motion, including all planetary perturbations or only those indicated as relevant by the developed 2D maps. It allowed validating these maps as a powerful tool.

- A comparison between two of the most popular special perturbation techniques, Cowell and Encke, demonstrated that round off errors take its toll even working in double precision and showed that Encke's method performs much better due to its mathematical foundations since it computes δr that presumably changes more slowly than r does, thus allowing larger step sizes to be taken

- A simple debugging test carried out by MatLab showed that both time and computational resources (amount of data) are sensibly reduced by not considering those negligible bodies suggested by the application of maps and tests (it would offer time saving in accessing and downloading ephemeris data from the platform while avoiding loading times to create large data matrices in the workspace)

Despite the outputs provided by the tool developed are a function of the type of propulsion on which a particular mission is based, the whole logical process followed from the beginning is universally valid and can therefore be replicated as many times as desired for different needs. Since cost and time are two of the most preponderant issues in our modern society, it is hoped that the procedure presented can make a small contribution and be improved in future developments.

Further development

Several actuations are suggested in prevision of being a possibility to improve this work:

- To promote a finer and deeper examination about whether various perturbation sources other than those gravitational could become important for interplanetary trajectories in general and for the BepiColombo's one in particular.

- To perform the propagation of interplanetary trajectories presented here using a low-thrust model, since it would be much more representative for the perturbing acceleration threshold estimated based on electric propulsion data

  In addition, in the dynamic version of propagation performed, when the spacecraft passes through areas where the threshold is not exceeded (and therefore the term perturbative in propagation is not considered) it is implicitly assumed that the spacecraft itself would be cancelling this vector resulting from perturbation in reality (since it would have sufficient propulsive capacity to do so). However, in the simulation this would need to be contemplated by the inclusion of an additional acceleration term in the perturbation model to oppose this disturbance, that is, by means of low - thrust propagation, which constitutes a step beyond in complexity.

- It is worth to carry out the numerical propagation exploring alternative schemes to Runge-Kutta, since there exists other that genuinely offer larger robustness for interplanetary trajectories.

A recommended one is, for instance, the Gauss-Jackson or Sum Squared method ([B.4],[15]) . This method – which is more complex to implement – is one of the best and most used for trajectory problems of the Cowell and Encke type. It is designed for the integration of systems of second order equations and is faster than integrating two first order equations. It also exhibits especially good control of accumulated round off errors.

- To apply the knowledge provided by the procedure developed and 2D maps to a real case, consisting of evaluating the feasibility of an asteroid rendezvous with a spacecraft departing from Earth

1) Lambert Arc is used (previous study of pork chops that show dates of departure/arrival or taking used ones for a mission realised in reality).

2) Propagation using the Arc with two bodies - 2BP Boundary value problem - and intercepts the asteroid on arrival date (simple model)

3) It is propagated using the NBP equations of motion with the dynamic propagator implemented and the output speed and flight time provided by the Arc after applying the differential corrector and the continuation method. A different point should be reached at the arrival date than the one occupied by the asteroid. That remoteness manifests:

- In terms of physical distance (km) Poor but more visually intuitive information
- ΔV necessary to correct (with Differential corrector) the departure speed from the Earth (at departure date) to intercept satisfactorily the position occupied by the asteroid at arrival date (to be added to the one already provided by the Arc itself).

4) If this distance is large - especially in terms of ΔV - it will mean that the planets responsible for the sections in which the threshold is exceeded have gravitationally influenced the movement of the ship in a decisive way (double check for the validity of 2D maps).

5) Ultimately, it allows us to know - applying the limitations imposed by the ESA margin document - whether or not the spacecraft is able to perform that mission (based on its propulsive characteristics). The following diagram shows the logical steps that would be followed.

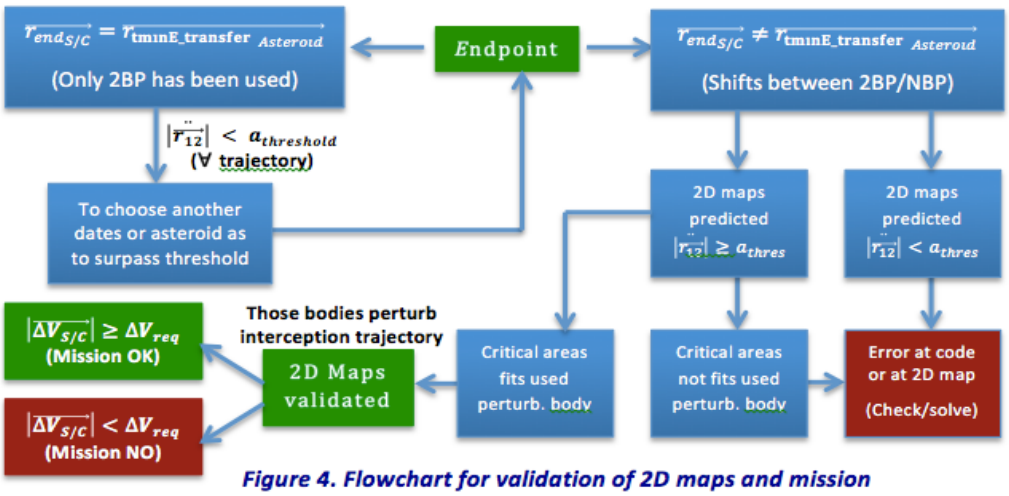

Figure 4. Flowchart for validation of 2D maps and mission

REFERENCES

[1] Malcolm Longair. *A history of astronomy, astrophysics and cosmology*. Encyclopedia of Life Support Systems (EOLSS).

[2] Grebow, D., *Generating periodic orbits in the circular restricted three-body problem with applications to Lunar South Pole coverage*, Purdue University, West Lafayette, Indiana, 2006.

[3] Parker, J. and Anderson, R. *Low – Energy Lunar trajectory design*. Jet Propulsion Laboratory. Pasadena, California (2013)

[4] Newhall, X. *Numerical representation of Planetary Ephemerides*. Jet Propulsion Laboratory. Celestial Mechanics and Dynamical Astronomy, Vol. 45, p. 305, 1989.

[5] HORIZONS platform. url: https://ssd.jpl.nasa.gov/?horizons_doc

[6] Diogene, A. *Dynamical substitutes of Lagrangian points and quasi-periodic orbits about them*. MSc in Space Engineering. Politecnico de Milano, July 2014.

[7] G. Gómez, A. Jorba, J. Masdemont, and C. Simó. *Study refinement of semi-analytical halo orbit theory*. Final Report, ESOC Contract, 1991

[8] Guojian Tang, Gerard Gomez Muntaner, Josep Masdemont Soler, L. Yijun, et al. *A note on the dynamics around the l1,2 lagrange points of the earthmoon system in a complete solar system model*. Springer Science+Business Media Dordrecht 2013, January 2012.

[9] Yijun Lian, Gerard Gómez, Josep J. Masdemont, and Guojian Tang. *A note on the dynamics around the lagrange collinear points of the earth– moon system in a complete solar system model*. Celestial Mechanics and Dynamical Astronomy, 115(2):185–211, 2013.

[10] Palacios, A. *Study and Application of Special Perturbation Methods*. School of Aerospace Engineering. Polythechnical University of Madrid, (2018)

[11] Newton, Sir Isaac. *Principia*. Motte's translation revised by Cajori. Vol. 1.Berkeley and Los Angeles, University of California Press, 1962.

[12] Bate, R., Mueller, D. and White, J. *Fundamentals of Astrodynamics*. Dover Publications, Inc.,1971.

[13] Townsend, G. E., Jr. *Orbital Flight Handbook*, Vol. 1, Part 1. NASA SP-33. National Aeronautics and Space Administration, Washington, DC, 1963.

[14] Vallado, David A. *An Analysis of State Vector Propagation using Differing Flight DynamicsPrograms*. Paper AAS 05-199 presented at the AAS/AIAA Space Flight Mechanics Conference. Copper Mountain, CO, 2005.

[15] Baker, R. and Makemson, M. *An Introductions to Astrodynamics*, 2nd ed. New York, Academic Press, 1967.

[16] William M. Folkner, James G. Williams, Dale H. Boggs, Ryan S. Park, and Petr Kuchynka. *The planetary and lunar ephemerides de430 and de431*. Technical report, JPL Interplanetary Network Progress Report 42-196, 2014.

[17] E. M. Standish. *JPL planetary and lunar ephemerides, de405/le405*. Interoffice memorandum IOM 312.F - 98 -84, Jet Propulsion Laboratory, August 26 1998.

[18] Herget, P. *The Computation of Orbits*. Privately printed. Edwards Brothers, Inc. Professor of Astronomy, University of Cincinnati, 1948.

[19] *Planetary Coordinates for the Years 1960 – 19880,* Her Majesty's Stationery Office, London.

[20] Chicarro, A & Martin, Patrick & Trautner, Roland. *The Mars Express mission: An overview*. European Space Agency, (Special Publication) ESA SP. 1240. 3-13, 2004.

[21] Hechler, Martin & Yáñez, Arturo. *Mars Express orbit design*. Acta Astronautica. 53. 497-507. 10.1016/S0094-5765(03)80010-3, 2003

[22] Estublier, D., Saccoccia, G. and González, J. *Electric Propulsion on SMART-1: A Technology Milestone.* ESA Bulletin 129 (Feb 2007).

[23] Garcia Yarnoz, D., Jehn, R. and Croon M. *Interplanetary navigation along the low-thrust trajectory of BepiColombo*. Acta Astronautica 59, 284 – 293, 2006

[24] G.D., Racca, G. P., Whitcomb and B. H., Foing. *The SMART-1 Mission*. ESA Bulletin 95, Aug 1998.

[25] Dumazert, P., Marchandise, F., Prioul, M. and Jolivet, L. *PPS® - 1350 –G Qualification Status May*. AIAA 2003 – 4549, 2003.

[26] Marchandise, F., Biron, J., Gambon, M. and Cornu, N. *The PPS®1350-G qualification demonstration: 7500 hrs on the ground and 5000 hrs in flight*. AIAA 2007 – 5197.

[27] Martin J.L Turner. *Rocket and Spacecraft Propulsion. Principles, Practice and New Developments*. Ed. Springer – Praxis, 2006.

[28] Kugelberg, J., Bodin, P., Persson, S. and Rathsman, P. *Accommodating electric propulsion on SMART-1*. Acta Astronautica 55, 121 – 130, 2004.

[29] Koppel, C. and Estublier, D. *The SMART-1 Hall Effect Thruster Around the Moon: In Flight Experience*. IEPC – 2005 – 119

[30] *Margin philosophy for science assessment studies*. ESA Unclassified document (ESTEC)

[31] R. A. Levis, J. Pérez, N. Coombs. *Qualification of the T6 Thruster for BepiColombo*. IEPC-2015-132／ISTS-2015-b-132

[32] "*Deep Space 1 Asteroid Flyby*" (PDF) (Press kit). NASA. 26 July 1999. Retrieved 20 November 2016

[33] Polk, Jay & Kakuda, R.Y. & Anderson, John & Brophy, John & Rawlin, V.K. & Sovey, J & Hamley, J. (2000). *Performance of the NSTAR ion propulsion system on the Deep Space One mission*. IEEE Aerospace Conference Proceedings. 4. 123 - 148 vol.4. 10.1109/AERO.2000.878373.

[34] https://www.nasa.gov/centers/glenn/about/history/ds1opseq.html

[35] Nishiyama, K., Hosoda,S., Ueno, K., Tsukizaki, R. and Kuninaka, H. *Development and Testing of the Hayabusa-2 Ion Engine System*. IEPC-2015-333／ISTS-2015-b-333

[36] C. Garner and M. Rayman, "*In-flight operation of the Dawn ion propulsion system: status at one year from the Vesta rendezvous*", 46th AIAA/ASME/SAE/ASEE Joint Propulsion Conference, Nashville, July 25-28, 2010

[37] https://nssdc.gsfc.nasa.gov/nmc/spacecraft/display.action?id=2007-043A

[38] https://solarsystem.nasa.gov/missions/dawn/overview/

[39] https://www.airbus.com/content/dam/stock-and-creative/infographic/BepiColombo-mission_Copyright%20Airbus%202018.pdf

[40] Conway, B.A. *Spacecraft Trajectory Optimization*. Cambridge University Press, Cambridge (2010)

[41] Bernelli, F., Vasile, M., Fornasari, N. and Masarati, P. *Design of Interplanetary and Lunar Missions Combining Low Thrust and Gravity Assists*. Final Report of ESA/ESOC Study Contract No. 14126/00/D/CS. Dipartamento di Ingegneria Aerospaziale. Politecnico di Milano (2002)

[42] Kemble, S. *Interplanetary Mission Analysis and Design*. Ed. Springer – Praxis (2006)

[43] Brouwer, D. and Clemence, G. M. Methods of Celestial Mechanics. New York, Academic Press, 1961.

BIBLIOGRAPHY

[B.1] Wiggins, S., *Introduction to Applied Nonlinear Dynamical Systems and Chaos* ,Springer, 2nd edition, 2003.

[B.2] Verhulst, F., *Nonlinear Differential Equations and Dynamical Systems*, 2nd edition, Universitext, 2000.

[B.3] Kemble, S. *Interplanetary Mission Analysis and Design*. Ed. Springer – Praxis (2006)

[B.4] Bate, R., Mueller, D.,*Fundamentals of Astrodynamics*. Dover Publications, Inc. 1971

[B.5] G.P, Sutton, O. Biblarz. *Rocket Propulsion Elements*. Ed. Wiley – Interscience (2001).

APPENDICES

APPENDIX A

Contained in this appendix, it is possible to find some additional results and/or analyses that are not strictly necessary to adequately follow the development of the thesis nor do they affect the fulfilment of the established objectives. They can be useful to deepen in some concepts, phenomena or results that can be interpreted in an alternative way, or simply out of curiosity (if it exists).

B.1 The Solar System Barycentre

It includes some pictures obtained from a simulation that allows observing the orbit of the Sun about the SSB, as well as to examine the relevance of each planetary effect. Their author is *Kirk Korista*, professor of Astronomy in the Department of Physics of the Western Michigan University and has been obtained from the URL:

http://homepages.wmich.edu/~korista/solarsystem_barycenter.pdf

Although it is convenient to think of the Sun as the stationary anchor of our solar system, it actually moves as the planets pull on it, causing it to orbit the barycentre of the solar system. The Sun never moves too far away from the barycentre of the solar system. The barycentre is often outside the Sun's photosphere, but never outside the Sun's corona. The simulation *ssbarycenter.gsim* allows observing the Sun's orbit about the solar system's barycentre. By removing the planets one by one, it is possible to observe the effect that each one has on the solar system's barycentre. Firstly, **Figure A-1** shows a picture of the Sun locked on the centre of the screen:

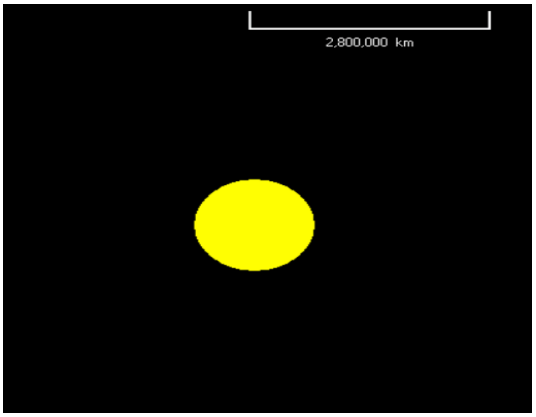

Figure A-1 The Sun and a reference for acquiring an order of magnitud about the scale sizes.

By editing the Sun and adjusting its size to 0 while maintaining its mass allows getting closer to the barycentre of the solar system and observing the trajectory of the Sun around it. It is shown at **Figure A-2**.

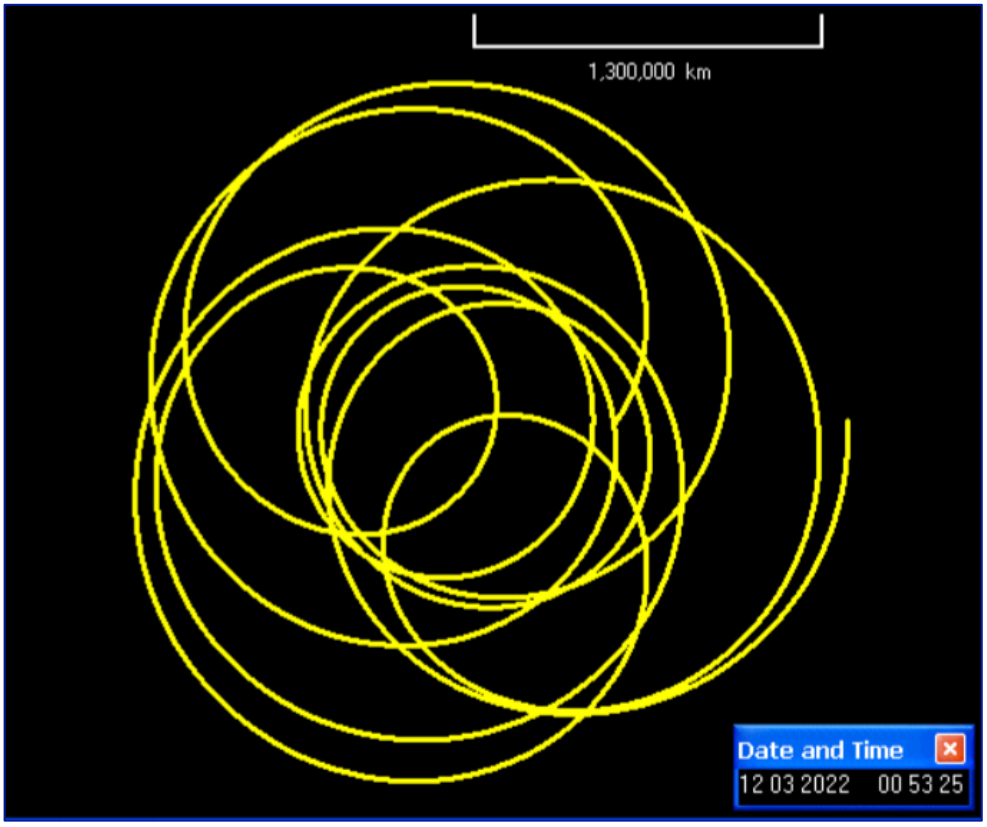

Figure A-2 This corresponds to the general motion (translational + wobbling) performed (and predicted) by the Sun from a given time in the past until March of 2022.

Editing Jupiter and setting its mass to 0 demonstrates that it is responsible for most of the wobbles. Setting then the mas of Saturn to 0, it is clear that it is the next strongest perturber as it can be seen in **Figure A-3**.

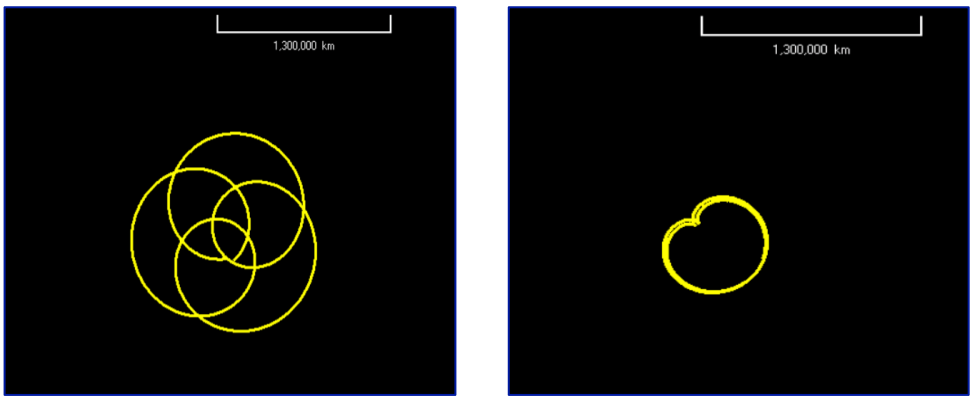

Figure A-3 Left) Removing Jupiter. Right) Removing Saturn as well.

Editing Neptune and setting its mass to 0 demonstrates that it the next strongest after Saturn. Uranus is responsible of this circle, since the Sun's period and Uranus period around the barycentre match. In addition, zooming in the picture exposes the effects of the smaller planets on the Uranus – induced circle. It can be seen in **Figure A-4.**

.

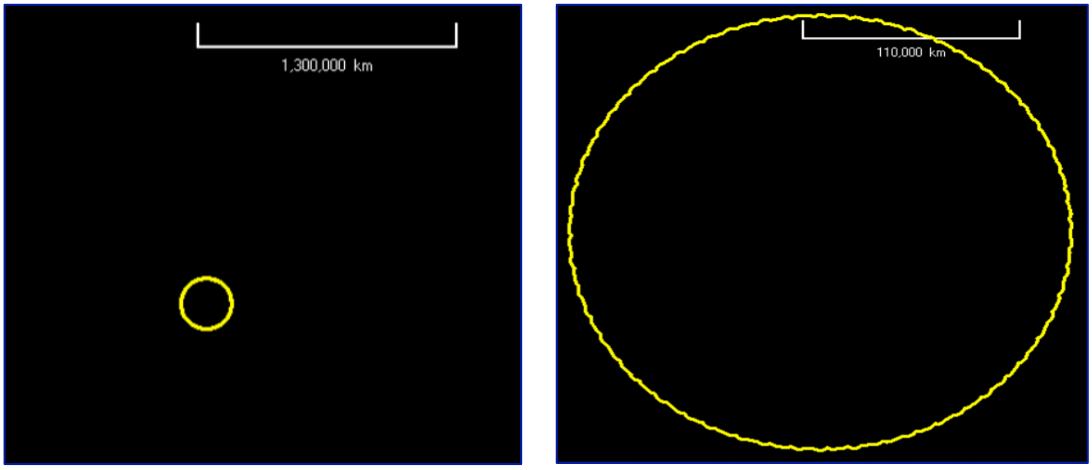

Figure A-4 The motion of Sun about the SSB once removed Jupiter + Saturn + Neptune. It indicates that the next massive planet (Uranus) is responsible for the shape of the circle (left). The remaining planets explain the oscillatons (right).

Setting now the mass of Uranus to 0 causes the Sun looking like apparently at rest with respect to SSB. However, zooming in on the picture shows the influence of the smallest remaining planets. The Earth – Moon system is now responsible for the majority of the wobble reflected in **Figure A-5**.

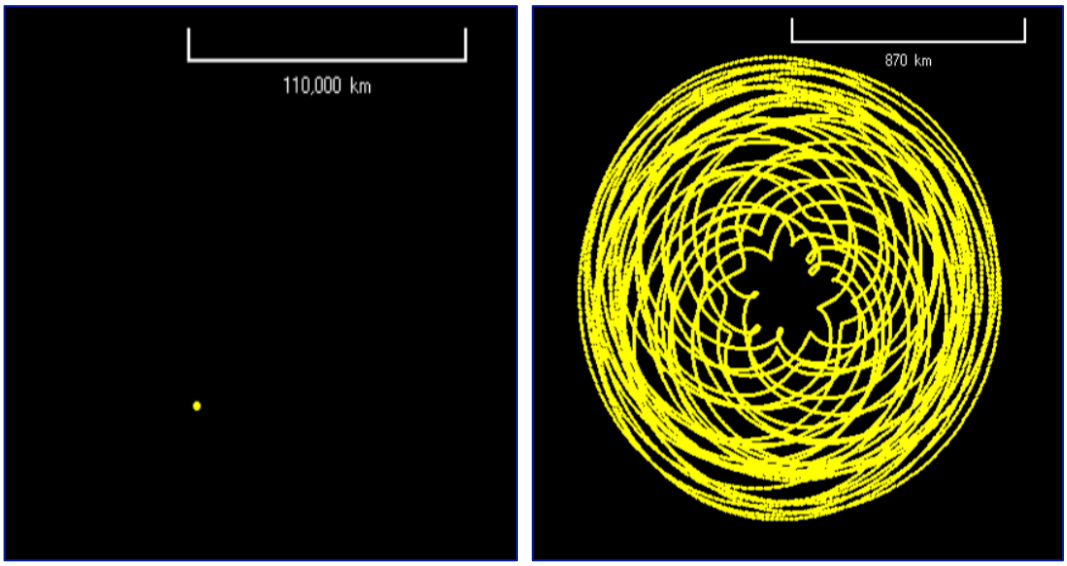

Figure A-5 The motion of Sun about SSB due to the Earth – Moon system.

Setting now the Earth – Moon mass to 0 leaves Venus as the next significant perturber, as shown in **Figure A-6**.

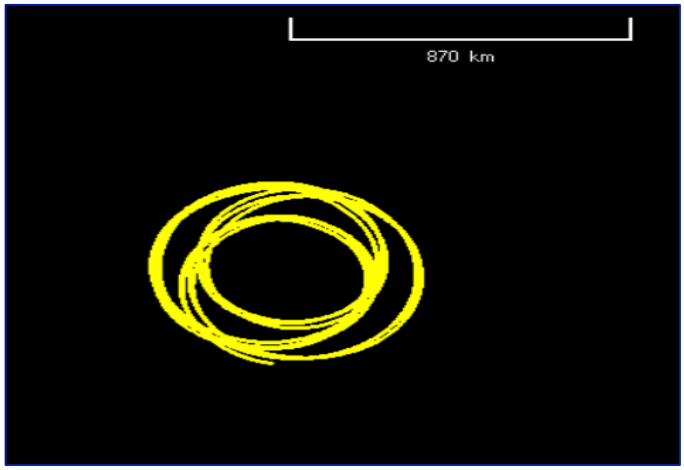

Figure A-6 The updated Sun motion about SSB once every mass has been removed except Mercury, Venus, Mars and Pluto.

Setting Venus mass to 0 leaves Mercury, Mars and Pluto as the only disturbances. Interestingly, the pattern of motion changes (zooming is clearer). Pluto's effect is just to move the pattern out of the centre due to its gravity tug.

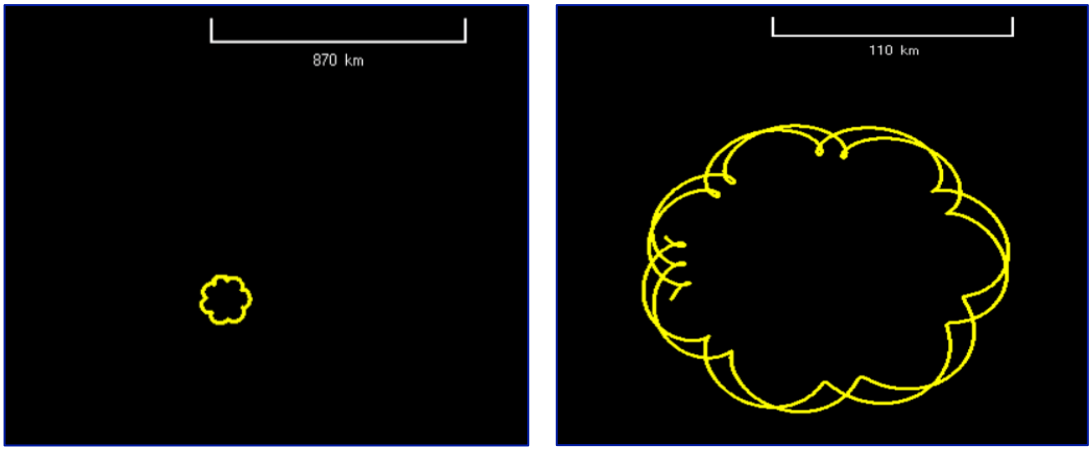

Figure A-7 The updated Sun motion about SSB once only Mercury, Mars and Pluto remain.

Setting Mars mass to 0 shows Mercury's influence causing the centre of the Sun to trace circles about SSB.

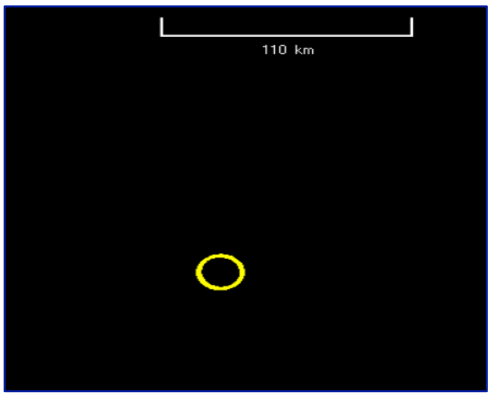

Figure A-8 The shape of Sun's motion about SSB when the "tug war" remains barely between Mercury and Pluto.

Finally, removing Mercury and letting the simulation run just for half of Pluto's orbit (due to its large period of almost 250 years) one can observe its unique influence on the Sun. Note the scale of motion (the Sun practically remains fixed).

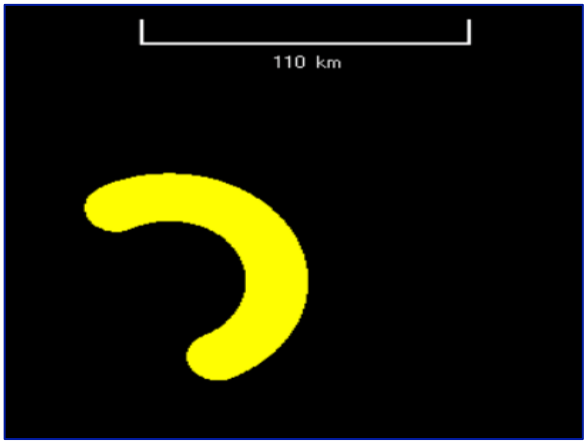

Figure A-9 The shape of Sun's motion influenced only by Pluto.

APPENDIX B Codes and algorithms

Here it will be included every *code*, *script* or *function* needed to support the calculations and results shown throughout the report. They will be presented in strict synchrony with the main body of the document.

A.1 Reading and loading of HORIZONS ephemerides

The following functions have been used to obtain all data (in MatLab):

• **Asteroid_cases.m** (Function that acts as a selector to choose the body object of the mission, being as options the asteroids *IRIS*, *QUETZALCOATL* and *ATEN* and the spacecraft *BepiColombo*)

• **Sun_Trajectory_HORIZONS.m** (To acquire Sun trajectory from HORIZONS)

• **Planet_Trajectory_HORIZONS.m** (To acquire Planet and Moon trajectories from HORIZONS)

• **Asteroid_Trajectory_HORIZONS.m** (To acquire asteroid and the S/C trajectories from HORIZONS)

Note: - The reason why three different categories of functions have been implemented respond to the fact that the structure of the .txt files varies quite between that of the Sun, the planets and other bodies (it was difficult to generalise). On the other hand, the reason why other bodies are considered in the selector function (three asteroids) lies in the fact that, before focussing on the main problem, many preliminary tests were carried out with them.

Asteroid_cases.m

```
function [r_Sun_SSB_eph,v_Sun_SSB_eph,r_Mercury_SSB_eph,v_Mercury_SSB_eph...
    ,r_Venus_SSB_eph,v_Venus_SSB_eph,r_Earth_SSB_eph,v_Earth_SSB_eph...
    ,r_Mars_SSB_eph,v_Mars_SSB_eph,r_Jupiter_SSB_eph,v_Jupiter_SSB_eph...
    ,r_Saturn_SSB_eph,v_Saturn_SSB_eph,r_Asteroid_SSB_eph,v_Asteroid_SSB_eph...
    ,r_Uranus_SSB_eph,v_Uranus_SSB_eph,r_Neptune_SSB_eph,v_Neptune_SSB_eph...
    ,r_Pluto_SSB_eph,v_Pluto_SSB_eph,r_Moon_SSB_eph,v_Moon_SSB_eph...
    ,threshold_exp] = Asteroid_cases(Asteroid,Propagation_time)
switch Asteroid
    case 'IRIS'
        if Propagation_time == 60 % [60 days]
        [r_Sun_SSB_eph,v_Sun_SSB_eph] = Sun_Trajectory_HORIZONS_60;
        [r_Asteroid_SSB_eph,v_Asteroid_SSB_eph] = Asteroid_Trajectory_HORIZONS_60;
        else % Asteroid Period (For IRIS 1347 days)
        [r_Sun_SSB_eph,v_Sun_SSB_eph] = Sun_Trajectory_HORIZONS_IRIS;
        [r_Mercury_SSB_eph,v_Mercury_SSB_eph] = Planet_Trajectory_HORIZONS_IRIS(1);
        [r_Venus_SSB_eph,v_Venus_SSB_eph] = Planet_Trajectory_HORIZONS_IRIS(2);
        [r_Earth_SSB_eph,v_Earth_SSB_eph] = Planet_Trajectory_HORIZONS_IRIS(3);
        [r_Mars_SSB_eph,v_Mars_SSB_eph] = Planet_Trajectory_HORIZONS_IRIS(4);
        [r_Jupiter_SSB_eph,v_Jupiter_SSB_eph] = Planet_Trajectory_HORIZONS_IRIS(5);
        [r_Saturn_SSB_eph,v_Saturn_SSB_eph] = Planet_Trajectory_HORIZONS_IRIS(6);
        [r_Asteroid_SSB_eph,v_Asteroid_SSB_eph] =
         Asteroid_Trajectory_HORIZONS_AP('IRIS',Propagation_time);
        threshold_exp = 0.5e6;
        end
        case 'QUETZALCOATL'
        if Propagation_time == 60 % [60 days]
        [r_Sun_SSB_eph,v_Sun_SSB_eph] = Sun_Trajectory_HORIZONS_60;
        [r_Asteroid_SSB_eph,v_Asteroid_SSB_eph] = Asteroid_Trajectory_HORIZONS_60;
```

```
        else % Asteroid Period (For QUETZALCOATL 1461 days)
        [r_Sun_SSB_eph,v_Sun_SSB_eph] = Sun_Trajectory_HORIZONS_QUETZALCOATL;
        [r_Mercury_SSB_eph,v_Mercury_SSB_eph] = Planet_Trajectory_HORIZONS_QUETZALCOATL(1);
         [r_Venus_SSB_eph,v_Venus_SSB_eph] = Planet_Trajectory_HORIZONS_QUETZALCOATL(2);
        [r_Earth_SSB_eph,v_Earth_SSB_eph] = Planet_Trajectory_HORIZONS_QUETZALCOATL(3);
        [r_Mars_SSB_eph,v_Mars_SSB_eph] = Planet_Trajectory_HORIZONS_QUETZALCOATL(4);
        [r_Jupiter_SSB_eph,v_Jupiter_SSB_eph] = Planet_Trajectory_HORIZONS_QUETZALCOATL(5);
        [r_Saturn_SSB_eph,v_Saturn_SSB_eph] = Planet_Trajectory_HORIZONS_QUETZALCOATL(6);
        [r_Asteroid_SSB_eph,v_Asteroid_SSB_eph] =
         Asteroid_Trajectory_HORIZONS_AP('QUETZALCOATL',Propagation_time);
        threshold_exp = 0.5e5;
        end
        case 'ATEN'
        if Propagation_time == 60 % [60 days]
        [r_Sun_SSB_eph,v_Sun_SSB_eph] = Sun_Trajectory_HORIZONS_60;
        [r_Asteroid_SSB_eph,v_Asteroid_SSB_eph] = Asteroid_Trajectory_HORIZONS_60;
        else % Asteroid Period (For 2062 ATEN 347 days)
        [r_Sun_SSB_eph,v_Sun_SSB_eph] = Sun_Trajectory_HORIZONS_ATEN;
        [r_Mercury_SSB_eph,v_Mercury_SSB_eph] = Planet_Trajectory_HORIZONS_ATEN(1);
        [r_Venus_SSB_eph,v_Venus_SSB_eph] = Planet_Trajectory_HORIZONS_ATEN(2);
        [r_Earth_SSB_eph,v_Earth_SSB_eph] = Planet_Trajectory_HORIZONS_ATEN(3);
        [r_Mars_SSB_eph,v_Mars_SSB_eph] = Planet_Trajectory_HORIZONS_ATEN(4);
        [r_Jupiter_SSB_eph,v_Jupiter_SSB_eph] = Planet_Trajectory_HORIZONS_ATEN(5);
        [r_Saturn_SSB_eph,v_Saturn_SSB_eph] = Planet_Trajectory_HORIZONS_ATEN(6);
        [r_Asteroid_SSB_eph,v_Asteroid_SSB_eph] =
         Asteroid_Trajectory_HORIZONS_AP('ATEN',Propagation_time);
        threshold_exp = 0.2e5;
        end
        case 'BepiColombo' % Spacecraft theoretical trajectory period (20/10/18 to 02/11/25) with
                           step 1 day (24 h) or 6h (it is changeable within its function)
        [r_Sun_SSB_eph,v_Sun_SSB_eph] = Sun_Trajectory_HORIZONS_BepiColombo;
        [r_Mercury_SSB_eph,v_Mercury_SSB_eph] = Planet_Trajectory_HORIZONS_BepiColombo(1);
        [r_Venus_SSB_eph,v_Venus_SSB_eph] = Planet_Trajectory_HORIZONS_BepiColombo(2);
        [r_Earth_SSB_eph,v_Earth_SSB_eph] = Planet_Trajectory_HORIZONS_BepiColombo(3);
        [r_Mars_SSB_eph,v_Mars_SSB_eph] = Planet_Trajectory_HORIZONS_BepiColombo(4);
        [r_Jupiter_SSB_eph,v_Jupiter_SSB_eph] = Planet_Trajectory_HORIZONS_BepiColombo(5);
        [r_Saturn_SSB_eph,v_Saturn_SSB_eph] = Planet_Trajectory_HORIZONS_BepiColombo(6);
        [r_Uranus_SSB_eph,v_Uranus_SSB_eph] = Planet_Trajectory_HORIZONS_BepiColombo(7);
        [r_Neptune_SSB_eph,v_Neptune_SSB_eph] = Planet_Trajectory_HORIZONS_BepiColombo(8);
        [r_Pluto_SSB_eph,v_Pluto_SSB_eph] = Planet_Trajectory_HORIZONS_BepiColombo(9);
        [r_Moon_SSB_eph,v_Moon_SSB_eph] = Planet_Trajectory_HORIZONS_BepiColombo(10);
        [r_Asteroid_SSB_eph,v_Asteroid_SSB_eph] =
         Asteroid_Trajectory_HORIZONS_AP('BepiColombo',Propagation_time);
end
end
```

Sun_Trajectory_HORIZONS.m

```
function [vector_positions,vector_velocities] = Sun_Trajectory_HORIZONS
%%% Acquisition of Sun trajectory from HORIZONS %%%
% Importation of file .txt given by the website tool:
load X_Sun_Bepi
clearvars -except X_Sun_Bepi Y_Sun_Bepi Z_Sun_Bepi
%% Generation of positions and velocities
time_Span = round(date2mjd2000([2025,11,2,8,42,0])-date2mjd2000([2018,10,20,2,13,0])); % [days]
vector_positions = zeros(time_Span,3);
vector_velocities = zeros(time_Span,3);
j = 1;
for i=50:4:41155
    % For coordinate X:
    pos_x_char = cell2mat(X_Sun_Bepi(i));
    pos_x_char = pos_x_char(4:end);
    pos_x_num = str2double(pos_x_char);
    vel_x_char = cell2mat(X_Sun_Bepi(i+1));
    vel_x_char = vel_x_char(4:end);
    vel_x_num = str2double(vel_x_char);
    % For coordinate Y:
    pos_y_char = cell2mat(Y_Sun_Bepi(i));
    pos_y_char = pos_y_char(4:end);
    pos_y_num = str2double(pos_y_char);
    vel_y_char = cell2mat(Y_Sun_Bepi(i+1));
    vel_y_char = vel_y_char(4:end);
    vel_y_num = str2double(vel_y_char);
    % For coordinate Z:
    pos_z_char = cell2mat(Z_Sun_Bepi(i));
    pos_z_char = pos_z_char(4:end);
```

```
    pos_z_num = str2double(pos_z_char);
    vel_z_char = cell2mat(Z_Sun_Bepi(i+1));
    vel_z_char = vel_z_char(4:end);
    vel_z_num = str2double(vel_z_char);
    % Updating the state vector:
    vector_positions(j,1) = pos_x_num; vector_positions(j,2) = pos_y_num; vector_positions(j,3) =
                pos_z_num;
    vector_velocities(j,1) = vel_x_num; vector_velocities(j,2) = vel_y_num; vector_velocities(j,3) =
                vel_z_num;
    j = j+1;
end
end
```

Planet_Trajectory_HORIZONS.m

```
function [vector_positions,vector_velocities] = Planet_Trajectory_HORIZONS(num)
%%% Acquisition of planet trajectory from HORIZONS %%%
% Importation of file .txt given by the website tool:
aux = 0;
switch num
    case 1
    load X_Mercury_Bepi
    clearvars -except X_Mercury_Bepi Y_Mercury_Bepi Z_Mercury_Bepi aux
    X = X_Mercury_Bepi;
    Y = Y_Mercury_Bepi;
    Z = Z_Mercury_Bepi;
    aux = 1;
    index1 = 51;
    index2 = 41156;
    case 2
    load X_Venus_Bepi
    clearvars -except X_Venus_Bepi Y_Venus_Bepi Z_Venus_Bepi aux
    X = X_Venus_Bepi;
    Y = Y_Venus_Bepi;
    Z = Z_Venus_Bepi;
    index1 = 52;
    index2 = 41157;
    case 3
    load X_Earth_Bepi
    clearvars -except X_Earth_Bepi Y_Earth_Bepi Z_Earth_Bepi aux
    X = X_Earth_Bepi;
    Y = Y_Earth_Bepi;
    Z = Z_Earth_Bepi;
    index1 = 59;
    index2 = 41164;
    case 4
    load X_Mars_Bepi
    clearvars -except X_Mars_Bepi Y_Mars_Bepi Z_Mars_Bepi aux
    X = X_Mars_Bepi;
    Y = Y_Mars_Bepi;
    Z = Z_Mars_Bepi;
    index1 = 53;
    index2 = 41158;
    case 5
    load X_Jupiter_Bepi
    clearvars -except X_Jupiter_Bepi Y_Jupiter_Bepi Z_Jupiter_Bepi aux
    X = X_Jupiter_Bepi;
    Y = Y_Jupiter_Bepi;
    Z = Z_Jupiter_Bepi;
    index1 = 53;
    index2 = 41158;
    case 6
    load X_Saturn_Bepi
    clearvars -except X_Saturn_Bepi Y_Saturn_Bepi Z_Saturn_Bepi aux
    X = X_Saturn_Bepi;
    Y = Y_Saturn_Bepi;
    Z = Z_Saturn_Bepi;
    index1 = 53;
    index2 = 41158;
    case 7
    load X_Uranus_Bepi
    clearvars -except X_Uranus_Bepi Y_Uranus_Bepi Z_Uranus_Bepi aux
    X = X_Uranus_Bepi;
    Y = Y_Uranus_Bepi;
    Z = Z_Uranus_Bepi;
    index1 = 53;
    index2 = 41158;
    case 8
    load X_Neptune_Bepi
    clearvars -except X_Neptune_Bepi Y_Neptune_Bepi Z_Neptune_Bepi aux
    X = X_Neptune_Bepi;
    Y = Y_Neptune_Bepi;
    Z = Z_Neptune_Bepi;
    index1 = 53;
    index2 = 41158;
    case 9
    load X_Pluto_Bepi
    clearvars -except X_Pluto_Bepi Y_Pluto_Bepi Z_Pluto_Bepi aux
    X = X_Pluto_Bepi;
```

```
    Y = Y_Pluto_Bepi;
    Z = Z_Pluto_Bepi;
    index1 = 48;
    index2 = 41153;
    case 10
    load X_Moon_Bepi
    clearvars -except X_Moon_Bepi Y_Moon_Bepi Z_Moon_Bepi aux
    X = X_Moon_Bepi;
    Y = Y_Moon_Bepi;
    Z = Z_Moon_Bepi;
    index1 = 53;
    index2 = 41158;
end
%% Generation of positions and velocities
time_Span = round(date2mjd2000([2025,11,2,8,42,0])-date2mjd2000([2018,10,20,2,13,0])); % [days]
vector_positions = zeros(time_Span,3);
vector_velocities = zeros(time_Span,3);
j = 1;
for i=index1:4:index2
    % For coordinate X:
    pos_x_char = cell2mat(X(i));
    pos_x_char = pos_x_char(4:end);
    pos_x_num = str2double(pos_x_char);
    vel_x_char = cell2mat(X(i+1));
    vel_x_char = vel_x_char(4:end);
    vel_x_num = str2double(vel_x_char);
    % For coordinate Y:
    pos_y_char = cell2mat(Y(i));
    pos_y_char = pos_y_char(4:end);
    pos_y_num = str2double(pos_y_char);
    vel_y_char = cell2mat(Y(i+1));
    vel_y_char = vel_y_char(4:end);
    vel_y_num = str2double(vel_y_char);
    % For coordinate Z:
    if aux == 0
    pos_z_char = cell2mat(Z(i));
    pos_z_char = pos_z_char(4:end);
    pos_z_num = str2double(pos_z_char);
    vel_z_char = cell2mat(Z(i+1));
    vel_z_char = vel_z_char(4:end);
    vel_z_num = str2double(vel_z_char);
    else
    pos_z_num = Z(i);
    vel_z_num = Z(i+1);
    end
    % Updating the state vector:
    vector_positions(j,1) = pos_x_num; vector_positions(j,2) = pos_y_num; vector_positions(j,3) =
        pos_z_num;
    vector_velocities(j,1) = vel_x_num; vector_velocities(j,2) = vel_y_num; vector_velocities(j,3) =
        vel_z_num;
    j = j+1;
end
end
```

Asteroid_Trajectory_HORIZONS.m

```
function [vector_positions,vector_velocities] = Asteroid_Trajectory_HORIZONS(name,Propagation_time)
%%% Acquisition of Asteroid trajectory from HORIZONS (IRIS, QUETZALCOATL, ATEN or BepiColombo) %%%
% Importation of file .txt given by the website tool:
spacecraft = 0;
switch name
    case 'IRIS'
    load X_IRIS_AP
    clearvars -except X_IRIS_AP Y_IRIS_AP Z_IRIS_AP Propagation_time spacecraft
    X = X_IRIS_AP;
    Y = Y_IRIS_AP;
    Z = Z_IRIS_AP;
    index1 = 66;
    index2 = 129378;
    case 'QUETZALCOATL'
    load X_QUETZALCOATL_AP
    clearvars -except X_QUETZALCOATL_AP Y_QUETZALCOATL_AP Z_QUETZALCOATL_AP Propagation_time
spacecraft
    X = X_QUETZALCOATL_AP;
    Y = Y_QUETZALCOATL_AP;
    Z = Z_QUETZALCOATL_AP;
    index1 = 66;
    index2 = 140323;
    case 'ATEN'
    load X_ATEN_AP
    clearvars -except X_ATEN_AP Y_ATEN_AP Z_ATEN_AP Propagation_time spacecraft
    X = X_ATEN_AP;
    Y = Y_ATEN_AP;
    Z = Z_ATEN_AP;
    index1 = 77;
    index2 = 33390;
    case 'BepiColombo'
    load X_Bepi
    clearvars -except X_Bepi Y_Bepi Z_Bepi Propagation_time spacecraft
```

```
    X = X_Bepi;
    Y = Y_Bepi;
    Z = Z_Bepi;
    index1 = 83;
    index2 = 41188; % NOTE: - To change when another "timing" for Bepi is selected
    spacecraft = 1;
end
%% Generation of positions and velocities
time_Span = Propagation_time; % [days]
if spacecraft == 1
   vector_positions = zeros(time_Span,3);
   vector_velocities = zeros(time_Span,3);
else
   vector_positions = zeros(24*time_Span+1,3);
   vector_velocities = zeros(24*time_Span+1,3);
end
j = 1;
for i=index1:4:index2
    % For coordinate X:
    pos_x_char = cell2mat(X(i));
    pos_x_char = pos_x_char(4:end);
    pos_x_num = str2double(pos_x_char);
    vel_x_char = cell2mat(X(i+1));
    vel_x_char = vel_x_char(4:end);
    vel_x_num = str2double(vel_x_char);
    % For coordinate Y:
    pos_y_char = cell2mat(Y(i));
    pos_y_char = pos_y_char(4:end);
    pos_y_num = str2double(pos_y_char);
    vel_y_char = cell2mat(Y(i+1));
    vel_y_char = vel_y_char(4:end);
    vel_y_num = str2double(vel_y_char);
    % For coordinate Z:
    pos_z_char = cell2mat(Z(i));
    pos_z_char = pos_z_char(4:end);
    pos_z_num = str2double(pos_z_char);
    vel_z_char = cell2mat(Z(i+1));
    vel_z_char = vel_z_char(4:end);
    vel_z_num = str2double(vel_z_char);
    % Updating the state vector:
    vector_positions(j,1) = pos_x_num; vector_positions(j,2) = pos_y_num; vector_positions(j,3) =
                pos_z_num;
    vector_velocities(j,1) = vel_x_num; vector_velocities(j,2) = vel_y_num; vector_velocities(j,3) =
                vel_z_num;
    j = j+1;
end
end
```

A.2 Implementation of Encke's method

Encke_method.m

```
function [x,rho_vector,dr,k,epsilon,flyby] = Encke_method(x,i,A_perturbation_X,A_perturbation_Y...
    ,A_perturbation_Z,rho,dr,options,muSun,r_Sun_SSB_eph,v_Sun_SSB_eph,rho_vector,k,epsilon,flyby)

simulation_time = 27000;      % Number of hours simulated of ephemeris trajectory (about 1094 days,
arbitrarily selected only for having those 3 types of flyby)
data_step = 24;               % Data separation time (from HORIZONS ephemeris)
tspan = 0:data_step*3600:(simulation_time-1)*3600;
% Transformation of position and velocity vectors from SSB to relative to Sun:
x(1:3) = x(1:3)-r_Sun_SSB_eph(i,:);
x(4:6) = x(4:6)-v_Sun_SSB_eph(i,:);
%% Possible rectification
% In case that the magnitude of the deviation vector grows too fast in
% comparison to how the magnitude of the osculating position vector does,
% it will be necessary to define a new osculating orbit:
if norm(dr(1:3))/norm(rho(1:3)) > epsilon
    rectification = 1;
    k = 1;
    epsilon = epsilon+0.00005;
else
    rectification = 0;
end
%% Initial conditions to start the method
if i == 1 || rectification == 1 || flyby == 1
   % A State matrix is gathered to univocally define the osculating orbit
   % (in order to avoid numerical truncation error):
   [r_ijk_vector,v_ijk_vector] = osculating_orbit_States(x,tspan,muSun);
   % At initial conditions, both the reference (osculating) orbit and the
   % perturbed orbit coincide. So, the initial conditions are the same for
   % both orbits:
   dr = [0 0 0 0 0 0]; % Deviation state vector
   rho_vector = [r_ijk_vector(:,:) v_ijk_vector(:,:)]; % State matrix (osculating orbit)
   rho = rho_vector(k,:); % State vector (osculating orbit)
```

```
    q_factor = 0;
    % For an integration step, calculate the deviation vector from the
    % reference (osculating) orbit, knowing position vector of the osculating
    % orbit, position vector of pertubed orbit and knowing that q(0) = 0:
    [~,dr] = ode45(@(t,dr)[dr(4);dr(5);dr(6);...
     A_perturbation_X+muSun/norm(rho(1:3))^3*(q_factor*x(1)-dr(1));... % X Component
     A_perturbation_Y+muSun/norm(rho(1:3))^3*(q_factor*x(2)-dr(2));... % Y Component
     A_perturbation_Z+muSun/norm(rho(1:3))^3*(q_factor*x(3)-dr(3))]... % Z Component
     ,tspan(i:i+1),dr,options);
     dr = dr(end,:);
     flyby = 0;
end
    %% Computation of intermediate states
    k = k+1;
    % New state of the osculating orbit:
    rho = rho_vector(k,:); % State vector (osculating orbit)
    % Now, it is time to compute q, which is an ancillary variable introduced
    % to solve the problem of difference of nearly equal quantities, based on
    % the states vector deviation and osculating orbit:
    q = -1/norm(rho(1:3))^2*(dr(1)*(rho(1)+1/2*dr(1))+dr(2)*(rho(2)+1/2*dr(2))...
        +dr(3)*(rho(3)+1/2*dr(3)));
    % Following, it is necessary to compute what we called "q factor", which is
    % the difference of nearly equal quantities. To do so, it is necessary to
    % expand the term in a binomial series:
    n = 100; % Number of terms of the expansion desired (as higher, more accurated will be the value)
    q_factor = 3*q;
    added_numerator = q_factor;
    for j=1:n
        added_numerator = added_numerator*q*(3+2*j);
        q_factor = q_factor+(-1)^j*added_numerator/factorial(j+1);
    end
%% Second integration of deviation acceleration
% The second integration must take place with the right step time, no
% matter if it is reached after the initialisation, after a rectification
% or in a normal execution:
if i == 1
    x(1:3) = rho(1:3)+dr(end,1:3); % Perturbed position vector
    x(4:6) = rho(4:6)+dr(end,4:6); % Perturbed velocity vector
end
[~,dr] = ode45(@(t,dr)[dr(4);dr(5);dr(6);...
A_perturbation_X+muSun/norm(rho(1:3))^3*(q_factor*x(1)-dr(1));... % X Component
A_perturbation_Y+muSun/norm(rho(1:3))^3*(q_factor*x(2)-dr(2));... % Y Component
A_perturbation_Z+muSun/norm(rho(1:3))^3*(q_factor*x(3)-dr(3))]... % Z Component
,tspan(i:i+1),dr,options);

dr = dr(end,:);
%% Update of perturbed position and velocity vectors
x(1:3) = rho(1:3)+dr(end,1:3); % Perturbed position vector
x(4:6) = rho(4:6)+dr(end,4:6); % Perturbed velocity vector
% Transformation of position and velocity vectors to SSB again:
x(end,1:3) = x(end,1:3)+r_Sun_SSB_eph(i+1,:);
x(end,4:6) = x(end,4:6)+v_Sun_SSB_eph(i+1,:);
end
```

osculating_orbit_States.m

```
function [r_ijk_vector,v_ijk_vector] = osculating_orbit_States(x,tspan,muSun)
%% Determination of the classical orbital elements from initial state expressed in classical inertial
coordinates
% Determination of the angular momentum vector from the known current
% state:
h = cross(x(1:3),x(4:6)); % [km^2/s]
% Determination of the eccentricity vector:
e = (cross(x(4:6),h)/muSun) - (x(1:3)/norm(x(1:3)));
% Determination of the semi-major axis:
a = norm(h)^2/(muSun*(1-norm(e)^2));
% Determination of the semi-latus rectum:
p = norm(h)^2/muSun;
% Determination of the node vector:
n = cross([0 0 1],h);
% Determination of the orbital inclination:
i = acos(h(3)/norm(h));
% Determination of the longitude of the ascending node:
if n(2) >= 0
    Omega = acos(n(1)/norm(n));
else
    Omega = 2*pi-acos(n(1)/norm(n));
end
% Determination of the argument of periapsis:
if e(3) >= 0
    w = acos(dot(n,e)/(norm(n)*norm(e)));
else
    w = 2*pi-acos(dot(n,e)/(norm(n)*norm(e)));
end
% Determination of the initial true anomaly:
if dot(x(1:3),x(4:6)) >= 0
    True_anomaly = acos(dot(e,x(1:3))/(norm(e)*norm(x(1:3))));
else
    True_anomaly = 2*pi-acos(dot(e,x(1:3))/(norm(e)*norm(x(1:3))));
end
```

```
% Definition of a vector containing true anomaly in function of time:
True_anomaly_vector = zeros(1,length(tspan));
True_anomaly_vector(1) = True_anomaly;
% Now, a temporary-dependent true anomaly is computed:
for j=2:length(True_anomaly_vector)
    % Checking of the type of orbit:
    if norm(e) < 1 && norm(e) > 0 % Elliptic case
        % Determination of eccentric anomaly from true anomaly:
       E = 2*atan(tan(True_anomaly/2)/sqrt((1+norm(e))/(1-norm(e))));
       % Determination of mean anomaly using Kepler's equation:
       M = E-norm(e)*sin(E);
       % Update of Mean anomaly for the following time:
       M = M+sqrt(muSun/a^3)*(tspan(j)-tspan(j-1));
       % Solving Kepler's equation using Newton - Raphson to know the updated
       % eccentric anomaly:
       E = KeplerEqSolver(M,norm(e));
       % Finally, the updated true anomaly can be calculated:
       True_anomaly = 2*atan2(sqrt(1+norm(e))*sin(E/2),sqrt(1-norm(e))*cos(E/2));
       True_anomaly_vector(j) = True_anomaly;
    elseif norm(e) == 1 % Parabolic case
        disp('Parabola')
    elseif norm(e) > 1 % Hyperbolic case
        disp('Hyperbola')
    end
end
% Known the true_anomaly in function of time, the distance to the central
% body can be computed just using the polar equation of a conic section:
r_ijk_vector = zeros(length(tspan),3);
v_ijk_vector = zeros(length(tspan),3);
for j=1:length(True_anomaly_vector)
    True_anomaly = True_anomaly_vector(j);
    [r_ijk,v_ijk,~,~] = orb2att(p,norm(e),i,Omega,w,True_anomaly,muSun);
    r_ijk_vector(j,:) = r_ijk';
    v_ijk_vector(j,:) = v_ijk';
end
```

KeplerEqSolver.m

```
function E = KeplerEqSolver(M,e)
%---------------------------------------------------------------
% Function that solves the Kepler's equation  M=E-esin(E) for a
% given mean anomaly M and eccentricity 0<e<1.
% Input: M (rad) and e,   Output: E (eccentric anomaly in rad).
%---------------------------------------------------------------
E=M+e*sin(M); Co=E-e*sin(E)-M;
while abs(Co) > 1.e-14
 Co=-(E-e*sin(E)-M)/(1-e*cos(E));
 E=E+Co;
end
end
```

orb2att.m

```
function [r_ijk,v_ijk,r_PQR,v_PQR] = orb2att(p,e,i,Omega,w,True_anomaly,muSun)
% IMPORTANT: - Use Canonical units
% Calculate r and v in the Perifocal System from the already known orbital
% parameters:
r = p/(1+e*cos(True_anomaly));
r_PQR = [r*cos(True_anomaly) r*sin(True_anomaly) 0]; % In the form (P,Q,R)
v_PQR = [-sqrt(muSun/p)*sin(True_anomaly) sqrt(muSun/p)*(e+cos(True_anomaly)) 0]; % In
the form (P,Q,R)
% Rotation matrix for transforming: °°OJO que falla cuando i = 0 o bien
% e=0 (o ambas cosas a la vez)!!
R = [cos(Omega)*cos(w)-sin(Omega)*sin(w)*cos(i) -cos(Omega)*sin(w)-
sin(Omega)*cos(w)*cos(i) sin(Omega)*sin(i);sin(Omega)*cos(w)+cos(Omega)*sin(w)*cos(i) -
sin(Omega)*sin(w)+cos(Omega)*cos(w)*cos(i) -cos(Omega)*sin(i);sin(w)*sin(i)
cos(w)*sin(i) cos(i)];
% Transformation from Perifocal Reference System to Geocentric equatorial
% Reference System:
% Perifocal System: r = (P,Q,R)
% Geocentric equatorial System: r = (i,j,k);
r_ijk = R*r_PQR';
v_ijk = R*v_PQR';
end
```

A.3 Low-thrust mission analyses

```
% Analysis mass term ratio (ref. SMART-1)
%% SMART-1
mp0 = 82;        % [kg]
mse0 = 288;      % [kg]
max_inc = 10000; % Maximum value for increment of mass [kg]
%% DS1
mp0DS1 = 72;        % [kg]
mse0DS1 = 414;      % [kg]
%% HAYABUSA
mp0H1 = 66.2;        % [kg]
mse0H1 = 443.8;      % [kg]
%% HAYABUSA-2
mp0H2 = 66.5;        % [kg]
mse0H2 = 542.5;      % [kg]
%% DAWN
mp0DAWN = 425;        % [kg]
mse0DAWN = 793;      % [kg]
%% BEPICOLOMBO
mp0BP = 580;        % [kg]
mse0BP = 3494;      % [kg]
%% Mass matrix
% It is interesting to build a matrix containing each type of mass:
mass_matrix = [mp0 mse0;mp0DS1 mse0DS1;mp0H1 mse0H1;mp0H2 mse0H2;mp0DAWN mse0DAWN;mp0BP
mse0BP];
%% Analysis for 1 kg increments
% Constant propellant, variable structure and/or powerplant mass:
inc_mse = zeros(6,max_inc);
mp_inc_mse = zeros(6,max_inc);
k = 0;
for i=1:6
    for j=1:max_inc
        inc_mse(i,j) = mass_matrix(i,2)+k; % +1kg unit variation
        mp_inc_mse(i,j) = mass_matrix(i,1); % Constant
        k = k+1;
    end
    k = 0;
end
% Constant structural+powerplant, variable propellant mass:
inc_mp = zeros(6,max_inc-1);
mse_inc_mp = zeros(6,max_inc);
for i=1:6
    for j=1:max_inc
        inc_mp(i,j) = mass_matrix(i,1)+k; % +1kg unit variation
        mse_inc_mp(i,j) = mass_matrix(i,2); % Constant
        k =k+1;
    end
    k = 0;
end
mass_term_inc_mse = zeros(6,max_inc);
mass_term_inc_mp = zeros(6,max_inc);
for j=1:6
    for i=1:max_inc
        mass_term_inc_mse(j,i) =
log((inc_mse(j,i)+mp_inc_mse(j,i))/inc_mse(j,i))/mp_inc_mse(j,i);
        mass_term_inc_mp(j,i) =
log((inc_mp(j,i)+mse_inc_mp(j,i))/mse_inc_mp(j,i))/inc_mp(j,i);
    end
end
% Because we need just to analyze how affects the increments of 1kg in each
% case, it is neccesary to normalize by centering both independent terms at
% zero:
for i=1:6
    inc_mse(i,:) = inc_mse(i,:)-mass_matrix(i,2);
    inc_mp(i,:) = inc_mp(i,:)-mass_matrix(i,1);
end
%% Plot of results
figure
hold on

for i=1:6
    plot(inc_mse(i,:),mass_term_inc_mse(i,:))
    plot(inc_mp(i,:),mass_term_inc_mp(i,:))
end
xlabel('\fontsize {18} Mass increment (kg)')
ylabel('\fontsize {18} Mass term ratio value (kg^-1)')
legend('\fontsize {18} Increase in (Structural+powerplant) mass','\fontsize{18}Increase
in propellant mass')
```

```
title({'\fontsize {24} Behaviour of the mass ratio term when increasing 1 kg'; 'on
(structure + powerplant) or on propellant mass'})
%% Analysis of divergence
% Computing the difference:
difference = mass_term_inc_mp-mass_term_inc_mse;
figure
hold on
for i=1:6
    plot(inc_mse(i,:),difference(i,:))
end
xlabel('\fontsize {18} Mass increment (kg)')
ylabel('\fontsize {18} Difference mass ratios (kg^-1)')
title({'\fontsize {24} Difference mass ratios between both cases for each mission (kg^-
1)'})
legend('\fontsize {18} SMART-1','\fontsize{18} DEEP SPACE-1','\fontsize{18} HAYABUSA-
1','\fontsize{18} HAYABUSA-2','\fontsize{18} DAWN','\fontsize{18} BEPICOLOMBO')
```

A.4 Mapping of the Control Volume representing the Solar System

The following functions have been used to obtain all data (in MatLab):

• **General_acceleration_thresholds_2d_V2.m** (Main code in charge of performing all the process)

• **Sun_Trajectory_HORIZONS.m** (To acquire Sun trajectory from HORIZONS)

• **Planet_Trajectory_HORIZONS.m** (To acquire Planet and Moon trajectories from HORIZONS)

• **Planetary_configuration.m** (To configure the plot representation suitably in case the main code is executed to test an individual planet, i.e, within its orbit and surroundings)

Note:- Sun_Trajectory_HORIZONS and Planet_Trajectory_HORIZONS have already been introduced in B.1.

General_acceleration_thresholds_2d_V2.m

```
%********************** 2D VERSION ***************************************

% We apply the N-body problem in order to compute the asteroid-to-Sun
% instantaneous relative acceleration when placed at a certain general
% position, taking into account the planets as perturbative bodies(so we do
% not need to integrate it).
%*******************************************************************************
%% Inputs
tic;
Planet = input('Of which planet do you want to compute its acceleration field?\n Key subindex:
1:Mercury 2:Venus\n 3:Earth 4:Mars 5:Jupiter 6:Saturn\n');
disp('Introduce which planets/bodies you want to be considered (note that the previous one must be
taken into account again)')
Mercury = input('Mercury? 1 = YES 0 = NO\n');
Venus = input('Venus? 1 = YES 0 = NO\n');
Earth = input('Earth? 1 = YES 0 = NO\n');
Mars = input('Mars? 1 = YES 0 = NO\n');
Jupiter = input('Jupiter? 1 = YES 0 = NO\n');
Saturn = input('Saturn? 1 = YES 0 = NO\n');
Moon = input('Moon? 1 = YES 0 = NO\n');
% Input to carry out analyses of worst scenarios between two consecutive
% planets (in order to check if their Perturbing Spheres intersect)
Study_Minimum_distance = input('Do you want to compute a 2D map of minimum distance? 1 = YES 0 =
NO');
Consecutive_planets = 0;
if Study_Minimum_distance == 1
    Consecutive_planets = input('Of which consecutive planets do you want the study of minimum
distance?\n 1: Mercury - Venus 2: Venus - Earth 3: Earth - Mars 4: Jupiter - Saturn');
end
%% Ancillary data (VERIFIED!)
AU2km = 149597870.700;          % 1AU [km]
G = 6.67408e-20;                % Universal gravitational constant [km^3/(kg.s^2)]
M_S = 1988500e24;               % Sun Mass [kg]
M_Mercury = 3.302e23*Mercury;   % Mercury's Mass [kg]
M_Venus = 48.685e23*Venus;      % Venus Mass [kg]
```

```
M_Earth = 5.97219e24*Earth;     % Earth's Mass [kg]
M_Mars = 6.4171e23*Mars;        % Mars Mass [kg]
M_Jupiter = 1898.13e24*Jupiter; % Jupiter Mass [kg]
M_Saturn = 5.6834e26*Saturn;    % Saturn Mass [kg]
M_Moon = 7.349e22*Moon;         % Moon Mass [kg]
%% Recovering the trajectory of the planet for drawing its exact orbit (Also for the Sun during its
period of time; VERIFIED!)
[r_Planet_SSB_eph,time_Span,v_Planet_SSB_eph] = Planet_Trajectory_HORIZONS(Planet); % (Verified)
[r_Sun_SSB_eph,v_Sun_SSB_eph] = Sun_Trajectory_HORIZONS(time_Span,Planet);          % (Verified)
% (Each planet trajectory is obtained from HORIZONS ephemerides from date
% 31th May of 2019 to the date corresponding which its sidereal period)
% NOTE: - Given that we just want an "arbitrary" date for establishing the
% planetary configuration (each fixed position with respect to the Sun) it
% is enough just storing the position of the Sun for the period of time
% corresponding to the considered planet sidereal orbit period.
% By the other hand, both trajectories of Sun and planet are referenced to
% the SSB coordinate frame, but we need a position referenced to the position
% of the Sun at everytime. Hence, the relative position is needed (it is
% important taking into account that, for gravitational purposes, the Sun
% must always remain at the origin of coordinates):
r_Planet_Sun_eph = r_Planet_SSB_eph-r_Sun_SSB_eph(1:length(r_Planet_SSB_eph),:);
v_Planet_Sun_eph = v_Planet_SSB_eph-v_Sun_SSB_eph(1:length(v_Planet_SSB_eph),:);
%% Computing the exact position of Sun and planets at an arbitrary date (VERIFIED!)
% The tested planet will occupy an arbitrary position for each test:
arbitrary_Date_planet = ceil(rand(1)*length(r_Planet_Sun_eph)); % [Row position]
rel_pos_planet_Sun = r_Planet_Sun_eph(arbitrary_Date_planet,:);  % Actual fixed position of the
tested planet
% The rest of planets will occupy an arbitrary position for each test as
% well (we load all the planet trajectories):
[r_Mercury_SSB_eph,time_Span_Mercury,v_Mercury_SSB_eph] = Planet_Trajectory_HORIZONS(1);
[r_Venus_SSB_eph,time_Span_Venus,v_Venus_SSB_eph] = Planet_Trajectory_HORIZONS(2);
[r_Earth_SSB_eph,time_Span_Earth,v_Earth_SSB_eph] = Planet_Trajectory_HORIZONS(3);
[r_Mars_SSB_eph,time_Span_Mars,v_Mars_SSB_eph] = Planet_Trajectory_HORIZONS(4);
[r_Jupiter_SSB_eph,time_Span_Jupiter,v_Jupiter_SSB_eph] = Planet_Trajectory_HORIZONS(5);
[r_Saturn_SSB_eph,time_Span_Saturn,v_Saturn_SSB_eph] = Planet_Trajectory_HORIZONS(6);
[r_Moon_SSB_eph,time_Span_Moon,v_Moon_SSB_eph] = Planet_Trajectory_HORIZONS(7);
% Note: - Moon's orbit takes place about the Earth. Because of that,r_Moon_SSB_eph contains the
% position of Moon for the same sidereal period than the Earth exhibits. It will imply the Moon
% orbiting about 13 times the Earth, but since it corresponds a trajectory centered in SSB, its
% position at a certain time will be pretty similar to the one exhibited by the Earth (only with
% small differences since it is actually orbiting about it)
% Markers to store the size of the matrix for each planet:
marker_Mercury = length(r_Mercury_SSB_eph);
marker_Venus = length(r_Venus_SSB_eph);
marker_Earth = length(r_Earth_SSB_eph);
marker_Mars = length(r_Mars_SSB_eph);
marker_Jupiter = length(r_Jupiter_SSB_eph);
marker_Saturn = length(r_Saturn_SSB_eph);
marker_Moon = length(r_Moon_SSB_eph); % Same as Earth
[r_Sun_SSB_eph_Mercury,v_Sun_SSB_eph_Mercury] = Sun_Trajectory_HORIZONS(time_Span_Mercury,1);
[r_Sun_SSB_eph_Venus,v_Sun_SSB_eph_Venus] = Sun_Trajectory_HORIZONS(time_Span_Venus,2);
[r_Sun_SSB_eph_Earth,v_Sun_SSB_eph_Earth] = Sun_Trajectory_HORIZONS(time_Span_Earth,3);
[r_Sun_SSB_eph_Mars,v_Sun_SSB_eph_Mars] = Sun_Trajectory_HORIZONS(time_Span_Mars,4);
[r_Sun_SSB_eph_Jupiter,v_Sun_SSB_eph_Jupiter] = Sun_Trajectory_HORIZONS(time_Span_Jupiter,5);
[r_Sun_SSB_eph_Saturn,v_Sun_SSB_eph_Saturn] = Sun_Trajectory_HORIZONS(time_Span_Saturn,6);
% NOTE: - Not necessary for the Moon since its stored trajectory corresponds to the own sidereal
% period of the Earth. Hence, Sun's trajectory for Moon's stored trajectory it is the same that for
Earth.
% As for the tested planet, the rest of them must be referenced to the
% position of the Sun, considered as the origin of a reference frame (again
% we compute for all planets in order to have all the needed data):
r_Mercury_Sun_eph = r_Mercury_SSB_eph-r_Sun_SSB_eph_Mercury;
v_Mercury_Sun_eph = v_Mercury_SSB_eph-v_Sun_SSB_eph_Mercury;
r_Venus_Sun_eph = r_Venus_SSB_eph-r_Sun_SSB_eph_Venus;
v_Venus_Sun_eph = v_Venus_SSB_eph-v_Sun_SSB_eph_Venus;
r_Earth_Sun_eph = r_Earth_SSB_eph-r_Sun_SSB_eph_Earth;
v_Earth_Sun_eph = v_Earth_SSB_eph-v_Sun_SSB_eph_Earth;
r_Mars_Sun_eph = r_Mars_SSB_eph-r_Sun_SSB_eph_Mars;
v_Mars_Sun_eph = v_Mars_SSB_eph-v_Sun_SSB_eph_Mars;
r_Jupiter_Sun_eph = r_Jupiter_SSB_eph-r_Sun_SSB_eph_Jupiter;
v_Jupiter_Sun_eph = v_Jupiter_SSB_eph-v_Sun_SSB_eph_Jupiter;
r_Saturn_Sun_eph = r_Saturn_SSB_eph-r_Sun_SSB_eph_Saturn;
v_Saturn_Sun_eph = v_Saturn_SSB_eph-v_Sun_SSB_eph_Saturn;
r_Moon_Sun_eph = r_Moon_SSB_eph-r_Sun_SSB_eph_Earth;
v_Moon_Sun_eph = v_Moon_SSB_eph-v_Sun_SSB_eph_Earth;
control = 0;
while control ==0
% Each test will lead to a different arbitrary date as to compute later the
% arbitrary positions when needed
if Planet == 1 % MERCURY
    arbitrary_Date_Mercury = arbitrary_Date_planet; % To avoid false computations in gravity section
else
    arbitrary_Date_Mercury = ceil(rand(1)*length(r_Mercury_SSB_eph)); % [Row position]
end
if Planet ==2 % VENUS
    arbitrary_Date_Venus = arbitrary_Date_planet; % To avoid false computations in gravity section
else
    arbitrary_Date_Venus = ceil(rand(1)*length(r_Venus_SSB_eph)); % [Row position]
end
if Planet ==3 % EARTH
    arbitrary_Date_Earth = arbitrary_Date_planet; % To avoid false computations in gravity section
```

```
else
    arbitrary_Date_Earth = ceil(rand(1)*length(r_Earth_SSB_eph)); % [Row position]
end
if Planet == 4 % MARS
    arbitrary_Date_Mars = arbitrary_Date_planet; % To avoid false computations in gravity section
else
    arbitrary_Date_Mars = ceil(rand(1)*length(r_Mars_SSB_eph)); % [Row position]
end
if Planet == 5 % JUPITER
    arbitrary_Date_Jupiter = arbitrary_Date_planet; % To avoid false computations in gravity section
else
    arbitrary_Date_Jupiter = ceil(rand(1)*length(r_Jupiter_SSB_eph)); % [Row position]
end
if Planet == 6 % SATURN
    arbitrary_Date_Saturn = arbitrary_Date_planet; % To avoid false computations in gravity section
else
    arbitrary_Date_Saturn = ceil(rand(1)*length(r_Saturn_SSB_eph)); % [Row position]
end
% And for the rest of planets (not accounting the tested one):
rel_pos_Mercury_Sun = r_Mercury_Sun_eph(arbitrary_Date_Mercury,:); % Actual fixed position of Mercury
rel_pos_Venus_Sun = r_Venus_Sun_eph(arbitrary_Date_Venus,:);       % Actual fixed position of Venus
rel_pos_Earth_Sun = r_Earth_Sun_eph(arbitrary_Date_Earth,:);       % Actual fixed position of Earth
rel_pos_Mars_Sun = r_Mars_Sun_eph(arbitrary_Date_Mars,:);          % Actual fixed position of Mars
rel_pos_Jupiter_Sun = r_Jupiter_Sun_eph(arbitrary_Date_Jupiter,:); % Actual fixed position of Jupiter
rel_pos_Saturn_Sun = r_Saturn_Sun_eph(arbitrary_Date_Saturn,:);    % Actual fixed position of Saturn
rel_pos_Moon_Sun = r_Moon_Sun_eph(arbitrary_Date_Earth,:);         % Actual fixed position of Moon
(same as Earth)
% Small block for activating the while loop that allows searching the
% minimum distance scenario between a given couple of consecutive planets:

if Consecutive_planets == 1
    Minimum_distance = norm(rel_pos_Mercury_Sun-rel_pos_Venus_Sun);
    dist = 3.9114e+07;
elseif Consecutive_planets == 2
    Minimum_distance = norm(rel_pos_Venus_Sun-rel_pos_Earth_Sun);
    dist = .9524e+07;
elseif Consecutive_planets == 3
    Minimum_distance = norm(rel_pos_Earth_Sun-rel_pos_Mars_Sun);
    dist = 5.5761e+07;
else
    Minimum_distance = norm(rel_pos_Jupiter_Sun-rel_pos_Saturn_Sun);
    dist = 5.6953e+08;
end

if Study_Minimum_distance == 1
   if Minimum_distance <= dist
      control = 1;
   end
end
end
% So, the position for Sun will be:
r_Sun = zeros(1,3); % The Sun occupies the origin of coordinates
%% Rotation about Z-axis to project about XY plane (for representation purposes)
% As we wish to map the acceleration values for all the discretized domain,
% the data cannot be 3D. Thus, it is necessary to project in XY as to make
% the Z coordinate as close as possible to zero (then we will just neglect
% them). The first we need is to compute the direction of angular momentum:
h =
cross(r_Planet_Sun_eph(round(length(r_Planet_Sun_eph)*rand(1)),:),v_Planet_Sun_eph(round(length(v_Pla
net_Sun_eph)*rand(1)),:));
inclination = acosd(h(3)/norm(h)); % It is always the same value (if well calculated, will be around
or its complementary 173∫ in Mercury case)
if inclination > 90
    inclination = 180-inclination;
end
% Now, for vary the inclination of the orbit and make the z-coordinate
% zero, we need first to know the line of nodes vector:
n = cross([0 0 1],h);
n = n/norm(n); % To make the vector unitary
if n(1)<0 || n(2)<0
    n = n*(-1);
end
% Finally, a rotation of inclination angle about the line of nodes axis:
Rotation_Matrix = [cosd(inclination)+n(1)^2*(1-cosd(inclination)) n(1)*n(2)*(1-cosd(inclination)) -
n(2)*sind(inclination);n(1)*n(2)*(1-cosd(inclination)) cosd(inclination)+n(2)^2*(1-cosd(inclination))
n(1)*sind(inclination);n(2)*sind(inclination) -n(1)*sind(inclination) cosd(inclination)];
for i=1:length(r_Planet_Sun_eph)
r_Planet_Sun_eph(i,:) = Rotation_Matrix*r_Planet_Sun_eph(i,:)';
end
% Now we can neglect the values of Z-coordinate (it is a good approximation):
r_Planet_Sun_eph(:,3) = 0;
rel_pos_planet_Sun = (Rotation_Matrix*rel_pos_planet_Sun(1,:)')';
rel_pos_planet_Sun(1,3) = 0;
%% Rotation about Z-axis to project about XY plane (for calculation purposes; VERIFIED!!
% We have to do now exactly the same for each of the rest of planets different to the tested one.
% The reason is based on the fact that later gravitational computations could lead to erroneous
% results in case that the planetary orbits and positions still remain in 3D.
r_Planets_Sun_eph = [r_Mercury_Sun_eph;r_Venus_Sun_eph;r_Earth_Sun_eph;r_Mars_Sun_eph;...
    r_Jupiter_Sun_eph;r_Saturn_Sun_eph;r_Moon_Sun_eph]; % Containing the info for all planets &
bodies
v_Planets_Sun_eph = [v_Mercury_Sun_eph;v_Venus_Sun_eph;v_Earth_Sun_eph;v_Mars_Sun_eph;...
```

```
    v_Jupiter_Sun_eph;v_Saturn_Sun_eph;v_Moon_Sun_eph]; % Containing the info for all planets &
bodies
rel_pos_Planets_Sun = [rel_pos_Mercury_Sun;rel_pos_Venus_Sun;rel_pos_Earth_Sun;...
    rel_pos_Mars_Sun;rel_pos_Jupiter_Sun;rel_pos_Saturn_Sun;rel_pos_Moon_Sun];

starter = 0; % In order to pick a position and velocity of the proper planet (not other)
k = 1;
Markers = zeros(1,7); % To properly locate each body trajectory within r_Planets_Sun_eph matrix
for j=1:7
[r,~,v] = Planet_Trajectory_HORIZONS(j);
h =
cross(r_Planets_Sun_eph(starter+round(length(r)*rand(1)),:),v_Planets_Sun_eph(starter+round(length(v)
*rand(1)),:));
starter = starter+length(r);
inclination = acosd(h(3)/norm(h)); % It is always the same value (if well calculated, will be around
7∫ or its complementary 173∫ in Mercury case)
alarm = 0;
if inclination > 90
   inclination = 180-inclination;
   alarm = 1;
end
% Now, for vary the inclination of the orbit and make the z-coordinate
% zero, we need first to know the line of nodes vector:
n = cross([0 0 1],h);
n = n/norm(n); % To make the vector unitary
if alarm ==1
   n = n*(-1);
end
% Finally, a rotation of inclination angle about the line of nodes axis:
Rotation_Matrix = [cosd(inclination)+n(1)^2*(1-cosd(inclination)) n(1)*n(2)*(1-cosd(inclination)) -
n(2)*sind(inclination);n(1)*n(2)*(1-cosd(inclination)) cosd(inclination)+n(2)^2*(1-cosd(inclination))
n(1)*sind(inclination);n(2)*sind(inclination) -n(1)*sind(inclination) cosd(inclination)];
rel_pos_Planets_Sun(j,:) = (Rotation_Matrix*rel_pos_Planets_Sun(j,:)')';
rel_pos_Planets_Sun(j,3) = 0;
    for i=1:length(r)
        r_Planets_Sun_eph(k,:) = (Rotation_Matrix*r_Planets_Sun_eph(k,:)')';
        r_Planets_Sun_eph(k,3) = 0;
        k = k+1;
    end
    Markers(1,j) = k-1;
end
%% Compute real orbit discretization (for calculations; VERIFIED!)
% The idea is to prepare a 2D window which contains the orbit we are
% dealing with. So, the computations of gravitational field can be carried
% on. For doing that, the window must be a cube which its dimension matches
% the maximum dimension of the orbit:
% X_grid = -10e7+min(r_Planet_Sun_eph(:,1)):1e5:max(r_Planet_Sun_eph(:,1))+10e7; % NOTA: - Bajar una
unidad el paso para mayor finura en el discretizado (para afinar cuantitativamente y verificar la
esfera de influencia. De momento dejarla asi, ya que para los estudios de "Presentacion 3" necesito
bajo coste computacional para obtener bastantes graficas)
% Y_grid = -10e7+min(r_Planet_Sun_eph(:,2)):1e5:max(r_Planet_Sun_eph(:,2))+10e7;
X_grid = -20e7+min(r_Planet_Sun_eph(:,1)):5e5:max(r_Planet_Sun_eph(:,1))+20e7; % NOTA: - Bajar una
unidad el paso para mayor finura en el discretizado (para afinar cuantitativamente y verificar la
esfera de influencia. De momento dejarla asi, ya que para los estudios de "Presentacion 3" necesito
bajo coste computacional para obtener bastantes graficas)
Y_grid = -20e7+min(r_Planet_Sun_eph(:,2)):5e5:max(r_Planet_Sun_eph(:,2))+20e7;
if length(Y_grid) > length(X_grid)
   X_grid = linspace(-
20e7+min(r_Planet_Sun_eph(:,1)),max(r_Planet_Sun_eph(:,1))+20e7,length(Y_grid));
else
   Y_grid = linspace(-
20e7+min(r_Planet_Sun_eph(:,2)),max(r_Planet_Sun_eph(:,2))+20e7,length(X_grid));
end
disp('Time until gravitational calculations starts was:')
toc
disp('seconds')
%% Gravitational acceleration field within a given elliptical orbit
% Computing accelerations at each position vector for the discretized circle:
tic
gravitational_accelerations = zeros(length(X_grid),length(Y_grid)); % Value of acceleration for each
position vector created before
B_vector = zeros(length(X_grid),length(Y_grid));                          % Value of perturbation
term for each position vector created before
ratio_B_A = zeros(length(X_grid),length(Y_grid));                  % Value of ratio for each
position vector created before
k = 1;
% Since positions of each planet are to be fixed for a performed test
% (planetary and tested planet configuration), it will be more efficient
% outside of the loop (the positions will not change):
% Key notation: 1 Sun 2 Grid position (asteroid/spacecraft) 3...8 Planets
r31 = -rel_pos_Planets_Sun(1,:); % Sun relative to Mercury
r41 = -rel_pos_Planets_Sun(2,:); % Sun relative to Venus
r51 = -rel_pos_Planets_Sun(3,:); % Sun relative to Earth
r61 = -rel_pos_Planets_Sun(4,:); % Sun relative to Mars
r71 = -rel_pos_Planets_Sun(5,:); % Sun relative to Jupiter
r81 = -rel_pos_Planets_Sun(6,:); % Sun relative to Saturn
r91 = -rel_pos_Planets_Sun(7,:); % Sun relative to Moon
% NOTE: - We have changed the signs because we previously defined each
% planetary position with respect the Sun (so, now we need the opposite vec
% tors)
for i=1:length(Y_grid)
```

```
    for j=1:length(X_grid)
    % First update each position vector: (Aquì no hace falta
    % condicionalidad porque el cuerpo 2 recuerda que es el asteroide).
    r12 = [X_grid(i) Y_grid(j) 0]; % Each position vector in the window (theoretical position
occupied by a generic asteroid/spacecraft wr Sun)
    r32 = r12+r31;                 % Asteroid relative to Mercury
    r42 = r12+r41;                 % Asteroid relative to Venus
    r52 = r12+r51;                 % Asteroid relative to Earth
    r62 = r12+r61;                 % Asteroid relative to Mars
    r72 = r12+r71;                 % Asteroid relative to Jupiter
    r82 = r12+r81;                 % Asteroid relative to Saturn
    r92 = r12+r91;                 % Asteroid relative to Moon
    % Computing direct gravitational term (A):
    A = -G*M_S/norm(r12)^3*r12; % (M_asteroid or M_spacecraft can be neglected in all cases)
    % Computing indirect gravitational term (B):
    B = -G*M_Mercury*(r32/norm(r32)^3-r31/norm(r31)^3)-G*M_Venus*(r42/norm(r42)^3-r41/norm(r41)^3)...
        -G*M_Earth*(r52/norm(r52)^3-r51/norm(r51)^3)-G*M_Mars*(r62/norm(r62)^3-r61/norm(r61)^3)...
        -G*M_Jupiter*(r72/norm(r72)^3-r71/norm(r71)^3)-G*M_Saturn*(r82/norm(r82)^3-
r81/norm(r81)^3)...
        -G*M_Moon*(r92/norm(r92)^3-r91/norm(r91)^3);
    % Update the relative acceleration:
    a12 = A+B; % [km/s]
    % Since we are no longer interested in the vector direction but in its
    % magnitude:
    a12 = norm(a12);
    gravitational_accelerations(j,i) = a12;
    B_vector(j,i) = norm(B);
    ratio_B_A(j,i) = norm(B)/norm(A);

    end
    k = k+3;
end
disp('Time spent in gravitational calculations was:')
toc
disp('seconds')
%% Plots and graphics
figure
hold on
contourf(X_grid,Y_grid,B_vector,min(min(B_vector)):2e-11:6.94e-10)
% PLANETS: (Mercury is out of the conditional sentences since it always must
% appear, whatever the chosen test planet be):
% MERCURY:
plot(r_Mercury_Sun_eph(:,1),r_Mercury_Sun_eph(:,2),'-r','LineWidth',0.5)
plot(rel_pos_Mercury_Sun(1),rel_pos_Mercury_Sun(2),'k.','MarkerSize',10)
% VENUS:
if Planet == 2 || Planet == 3 || Planet == 4 || Planet == 5 || Planet == 6
   plot(r_Venus_Sun_eph(:,1),r_Venus_Sun_eph(:,2),'-r','LineWidth',0.5)
   plot(rel_pos_Venus_Sun(1),rel_pos_Venus_Sun(2),'m.','MarkerSize',20)
end
% EARTH & MOON & MARS:
if Planet == 3 || Planet == 4 || Planet == 5 || Planet == 6
   plot(r_Earth_Sun_eph(:,1),r_Earth_Sun_eph(:,2),'-r','LineWidth',0.5)
   plot(rel_pos_Earth_Sun(1),rel_pos_Earth_Sun(2),'b.','MarkerSize',22)
   plot(rel_pos_Moon_Sun(1),rel_pos_Moon_Sun(2),'b.','MarkerSize',8)
   plot(r_Mars_Sun_eph(:,1),r_Mars_Sun_eph(:,2),'-r','LineWidth',0.5)
   plot(rel_pos_Mars_Sun(1),rel_pos_Mars_Sun(2),'r.','MarkerSize',15)
end
% JUPITER:

if Planet == 5 || Planet == 6
   plot(r_Jupiter_Sun_eph(:,1),r_Jupiter_Sun_eph(:,2),'-r','LineWidth',0.5)
   plot(rel_pos_Jupiter_Sun(1),rel_pos_Jupiter_Sun(2),'.','Color',[0.6 0.6 0.6],'MarkerSize',100)
end
% SATURN:

if Planet == 6
   plot(r_Saturn_Sun_eph(:,1),r_Saturn_Sun_eph(:,2),'-r','LineWidth',0.5)
   plot(rel_pos_Saturn_Sun(1),rel_pos_Saturn_Sun(2),'.','Color',[0.6 0.6 0.6],'MarkerSize',80)
end
xlabel('\fontsize {20} X Position vector (km)')
ylabel('\fontsize {20} Y Position vector (km)')
title('\fontsize {24} Perturbing term/(Sun-body) for Window 1 (S-M-V-E-M). Threshold: B >= 6.94e-10
km/s^2')
legend('\fontsize {20} Perturbing term (B) [km/s^2]','\fontsize {20} Mercury orbit','\fontsize {20}
Mercury','\fontsize {20} Venus orbit','\fontsize {20} Venus','\fontsize {20} Earth orbit','\fontsize
{20} Earth','\fontsize {20} Moon','\fontsize {20} Mars orbit','\fontsize {20} Mars','\fontsize {20}
Jupiter orbit','\fontsize {20} Jupiter','\fontsize {20} Saturn orbit','\fontsize {20} Saturn')
%% Planetary configuration (for comparison)
% Planetary trajectories (one orbit period for each):
Planetary_configuration(Planet,r_Sun,r_Planet_Sun_eph,r_Planets_Sun_eph,Markers,...
rel_pos_planet_Sun,rel_pos_Planets_Sun,Mercury,Venus,Earth,Mars,Jupiter,Saturn,Moon)
toc;
```

Planetary_configuration.m

```
function
Planetary_configuration(Planet,r_Sun,r_Planet_Sun_eph,r_Planets_Sun_eph,Markers,rel_pos_planet_Sun,re
l_pos_Planets_Sun,Mercury,Venus,Earth,Mars,Jupiter,Saturn,Moon)
% NOTA: - He tenido que hacerlo de esta forma tan "fea" y redundante porque
% si no MatLab no me ploteaba todo en una misma figura (cosas de MatLab):
```

```
if Planet == 1
% Exact position in the randomly tested scenario: (Sun is the origin of
% coordinates, placed at (0,0)):
figure
hold on
% SUN
plot3(r_Sun(1,1),r_Sun(1,2),r_Sun(1,3),'ok','MarkerFaceColor','y','MarkerSize',40)
% TESTED PLANET:
plot3(r_Planet_Sun_eph(:,1),r_Planet_Sun_eph(:,2),r_Planet_Sun_eph(:,3))
plot3(rel_pos_planet_Sun(1),rel_pos_planet_Sun(2),rel_pos_planet_Sun(3),'.k','MarkerSize',20)
% VENUS:
plot3(r_Planets_Sun_eph(Markers(1)+1:Markers(2),1),r_Planets_Sun_eph(Markers(1)+1:Markers(2),2),r_Pla
nets_Sun_eph(Markers(1)+1:Markers(2),3))
if Venus == 1
plot3(rel_pos_Planets_Sun(2,1),rel_pos_Planets_Sun(2,2),rel_pos_Planets_Sun(2,3),'m.','MarkerSize',26
)
end
% EARTH:
plot3(r_Planets_Sun_eph(Markers(2)+2:Markers(3),1),r_Planets_Sun_eph(Markers(2)+2:Markers(3),2),r_Pla
nets_Sun_eph(Markers(2)+2:Markers(3),3))
if Earth == 1
plot3(rel_pos_Planets_Sun(3,1),rel_pos_Planets_Sun(3,2),rel_pos_Planets_Sun(3,3),'b.','MarkerSize',28
)
end
% MARS:
plot3(r_Planets_Sun_eph(Markers(3)+1:Markers(4),1),r_Planets_Sun_eph(Markers(3)+1:Markers(4),2),r_Pla
nets_Sun_eph(Markers(3)+1:Markers(4),3))
if Mars == 1
plot3(rel_pos_Planets_Sun(4,1),rel_pos_Planets_Sun(4,2),rel_pos_Planets_Sun(4,3),'r.','MarkerSize',22
)
end
% JUPITER:
plot3(r_Planets_Sun_eph(Markers(4)+1:Markers(5),1),r_Planets_Sun_eph(Markers(4)+1:Markers(5),2),r_Pla
nets_Sun_eph(Markers(4)+1:Markers(5),3))
if Jupiter == 1
plot3(rel_pos_Planets_Sun(5,1),rel_pos_Planets_Sun(5,2),rel_pos_Planets_Sun(5,3),'.k','MarkerSize',12
0)
end
% SATURN:
plot3(r_Planets_Sun_eph(Markers(5)+1:Markers(6),1),r_Planets_Sun_eph(Markers(5)+1:Markers(6),2),r_Pla
nets_Sun_eph(Markers(5)+1:Markers(6),3))
if Saturn == 1
plot3(rel_pos_Planets_Sun(6,1),rel_pos_Planets_Sun(6,2),rel_pos_Planets_Sun(6,3),'k.','MarkerSize',10
0)
end
% MOON:
if Moon == 1
plot3(rel_pos_Planets_Sun(7,1),rel_pos_Planets_Sun(7,2),rel_pos_Planets_Sun(7,3),'k.','MarkerSize',15
)
end

axis equal
elseif Planet == 2
figure
hold on
% SUN:
plot3(r_Sun(1,1),r_Sun(1,2),r_Sun(1,3),'ok','MarkerFaceColor','y','MarkerSize',40)
% TESTED PLANET:
plot3(r_Planet_Sun_eph(:,1),r_Planet_Sun_eph(:,2),r_Planet_Sun_eph(:,3))
plot3(rel_pos_planet_Sun(1),rel_pos_planet_Sun(2),rel_pos_planet_Sun(3),'.k','MarkerSize',20)
% MERCURY:
plot3(r_Planets_Sun_eph(1:Markers(1),1),r_Planets_Sun_eph(1:Markers(1),2),r_Planets_Sun_eph(1:Markers
(1),3))
if Mercury == 1
plot3(rel_pos_Planets_Sun(1,1),rel_pos_Planets_Sun(1,2),rel_pos_Planets_Sun(1,3),'k.')
end
% EARTH:
plot3(r_Planets_Sun_eph(Markers(2)+2:Markers(3),1),r_Planets_Sun_eph(Markers(2)+2:Markers(3),2),r_Pla
nets_Sun_eph(Markers(2)+2:Markers(3),3))
if Earth == 1
plot3(rel_pos_Planets_Sun(3,1),rel_pos_Planets_Sun(3,2),rel_pos_Planets_Sun(3,3),'b.','MarkerSize',28
)
end
% MARS:
plot3(r_Planets_Sun_eph(Markers(3)+1:Markers(4),1),r_Planets_Sun_eph(Markers(3)+1:Markers(4),2),r_Pla
nets_Sun_eph(Markers(3)+1:Markers(4),3))
if Mars == 1
plot3(rel_pos_Planets_Sun(4,1),rel_pos_Planets_Sun(4,2),rel_pos_Planets_Sun(4,3),'r.','MarkerSize',22
)
end
% JUPITER:
plot3(r_Planets_Sun_eph(Markers(4)+1:Markers(5),1),r_Planets_Sun_eph(Markers(4)+1:Markers(5),2),r_Pla
nets_Sun_eph(Markers(4)+1:Markers(5),3))
if Jupiter == 1
plot3(rel_pos_Planets_Sun(5,1),rel_pos_Planets_Sun(5,2),rel_pos_Planets_Sun(5,3),'.k','MarkerSize',12
0)
end
% SATURN:
plot3(r_Planets_Sun_eph(Markers(5)+1:Markers(6),1),r_Planets_Sun_eph(Markers(5)+1:Markers(6),2),r_Pla
nets_Sun_eph(Markers(5)+1:Markers(6),3))
```

```
if Saturn == 1
plot3(rel_pos_Planets_Sun(6,1),rel_pos_Planets_Sun(6,2),rel_pos_Planets_Sun(6,3),'k.','MarkerSize',10
0)
end
% MOON:
if Moon == 1
plot3(rel_pos_Planets_Sun(7,1),rel_pos_Planets_Sun(7,2),rel_pos_Planets_Sun(7,3),'k.','MarkerSize',15
)
end
axis equal

elseif Planet ==
figure
hold on
% SUN:
plot3(r_Sun(1,1),r_Sun(1,2),r_Sun(1,3),'ok','MarkerFaceColor','y','MarkerSize',40)
% TESTED PLANET:
plot3(r_Planet_Sun_eph(:,1),r_Planet_Sun_eph(:,2),r_Planet_Sun_eph(:,3))
plot3(rel_pos_planet_Sun(1),rel_pos_planet_Sun(2),rel_pos_planet_Sun(3),'.b','MarkerSize',28)
% MERCURY:
plot3(r_Planets_Sun_eph(1:Markers(1),1),r_Planets_Sun_eph(1:Markers(1),2),r_Planets_Sun_eph(1:Markers
(1),3))
if Mercury == 1
plot3(rel_pos_Planets_Sun(1,1),rel_pos_Planets_Sun(1,2),rel_pos_Planets_Sun(1,3),'k.')
end
% VENUS:
plot3(r_Planets_Sun_eph(Markers(1)+1:Markers(2),1),r_Planets_Sun_eph(Markers(1)+1:Markers(2),2),r_Pla
nets_Sun_eph(Markers(1)+1:Markers(2),3))
if Venus == 1
plot3(rel_pos_Planets_Sun(2,1),rel_pos_Planets_Sun(2,2),rel_pos_Planets_Sun(2,3),'m.','MarkerSize',26
)
end
% MARS:
plot3(r_Planets_Sun_eph(Markers(3)+1:Markers(4),1),r_Planets_Sun_eph(Markers(3)+1:Markers(4),2),r_Pla
nets_Sun_eph(Markers(3)+1:Markers(4),3))
if Mars == 1
plot3(rel_pos_Planets_Sun(4,1),rel_pos_Planets_Sun(4,2),rel_pos_Planets_Sun(4,3),'r.','MarkerSize',22
)
end
% JUPITER:
plot3(r_Planets_Sun_eph(Markers(4)+1:Markers(5),1),r_Planets_Sun_eph(Markers(4)+1:Markers(5),2),r_Pla
nets_Sun_eph(Markers(4)+1:Markers(5),3))
if Jupiter == 1
plot3(rel_pos_Planets_Sun(5,1),rel_pos_Planets_Sun(5,2),rel_pos_Planets_Sun(5,3),'.k','MarkerSize',12
0)
end
% SATURN:
plot3(r_Planets_Sun_eph(Markers(5)+1:Markers(6),1),r_Planets_Sun_eph(Markers(5)+1:Markers(6),2),r_Pla
nets_Sun_eph(Markers(5)+1:Markers(6),3))
if Saturn == 1
plot3(rel_pos_Planets_Sun(6,1),rel_pos_Planets_Sun(6,2),rel_pos_Planets_Sun(6,3),'k.','MarkerSize',10
0)
end
% MOON:
if Moon == 1
plot3(rel_pos_Planets_Sun(7,1),rel_pos_Planets_Sun(7,2),rel_pos_Planets_Sun(7,3),'k.','MarkerSize',15
)
end
axis equal
elseif Planet == 4
figure
hold on

% SUN:
plot3(r_Sun(1,1),r_Sun(1,2),r_Sun(1,3),'ok','MarkerFaceColor','y','MarkerSize',40)
% TESTED PLANET:
plot3(r_Planet_Sun_eph(:,1),r_Planet_Sun_eph(:,2),r_Planet_Sun_eph(:,3))
plot3(rel_pos_planet_Sun(1),rel_pos_planet_Sun(2),rel_pos_planet_Sun(3),'.r','MarkerSize',22)
% MERCURY:
plot3(r_Planets_Sun_eph(1:Markers(1),1),r_Planets_Sun_eph(1:Markers(1),2),r_Planets_Sun_eph(1:Markers
(1),3))
if Mercury == 1
plot3(rel_pos_Planets_Sun(1,1),rel_pos_Planets_Sun(1,2),rel_pos_Planets_Sun(1,3),'k.')
end
% VENUS:
plot3(r_Planets_Sun_eph(Markers(1)+1:Markers(2),1),r_Planets_Sun_eph(Markers(1)+1:Markers(2),2),r_Pla
nets_Sun_eph(Markers(1)+1:Markers(2),3))
if Venus == 1
plot3(rel_pos_Planets_Sun(2,1),rel_pos_Planets_Sun(2,2),rel_pos_Planets_Sun(2,3),'m.','MarkerSize',26
)
end
% EARTH:
plot3(r_Planets_Sun_eph(Markers(2)+2:Markers(3),1),r_Planets_Sun_eph(Markers(2)+2:Markers(3),2),r_Pla
nets_Sun_eph(Markers(2)+2:Markers(3),3))
if Earth == 1
plot3(rel_pos_Planets_Sun(3,1),rel_pos_Planets_Sun(3,2),rel_pos_Planets_Sun(3,3),'b.','MarkerSize',28
)
end
% JUPITER:
plot3(r_Planets_Sun_eph(Markers(4)+1:Markers(5),1),r_Planets_Sun_eph(Markers(4)+1:Markers(5),2),r_Pla
nets_Sun_eph(Markers(4)+1:Markers(5),3))
```

```
if Jupiter == 1
plot3(rel_pos_Planets_Sun(5,1),rel_pos_Planets_Sun(5,2),rel_pos_Planets_Sun(5,3),'.k','MarkerSize',12
0)
end
% SATURN:
plot3(r_Planets_Sun_eph(Markers(5)+1:Markers(6),1),r_Planets_Sun_eph(Markers(5)+1:Markers(6),2),r_Pla
nets_Sun_eph(Markers(5)+1:Markers(6),3))
if Saturn == 1
plot3(rel_pos_Planets_Sun(6,1),rel_pos_Planets_Sun(6,2),rel_pos_Planets_Sun(6,3),'k.','MarkerSize',10
0)
end
% MOON:
if Moon == 1
plot3(rel_pos_Planets_Sun(7,1),rel_pos_Planets_Sun(7,2),rel_pos_Planets_Sun(7,3),'k.','MarkerSize',15
)
end
axis equal
elseif Planet == 5
figure
hold on
% SUN:

plot3(r_Sun(1,1),r_Sun(1,2),r_Sun(1,3),'ok','MarkerFaceColor','y','MarkerSize',10)
% TESTED PLANET:
plot3(r_Planet_Sun_eph(:,1),r_Planet_Sun_eph(:,2),r_Planet_Sun_eph(:,3))
plot3(rel_pos_planet_Sun(1),rel_pos_planet_Sun(2),rel_pos_planet_Sun(3),'.k','MarkerSize',120)
% MERCURY:
plot3(r_Planets_Sun_eph(1:Markers(1),1),r_Planets_Sun_eph(1:Markers(1),2),r_Planets_Sun_eph(1:Markers
(1),3))
if Mercury == 1
plot3(rel_pos_Planets_Sun(1,1),rel_pos_Planets_Sun(1,2),rel_pos_Planets_Sun(1,3),'k.')
end
% VENUS:
plot3(r_Planets_Sun_eph(Markers(1)+1:Markers(2),1),r_Planets_Sun_eph(Markers(1)+1:Markers(2),2),r_Pla
nets_Sun_eph(Markers(1)+1:Markers(2),3))
if Venus == 1
plot3(rel_pos_Planets_Sun(2,1),rel_pos_Planets_Sun(2,2),rel_pos_Planets_Sun(2,3),'m.','MarkerSize',26
)
end
% EARTH:
plot3(r_Planets_Sun_eph(Markers(2)+2:Markers(3),1),r_Planets_Sun_eph(Markers(2)+2:Markers(3),2),r_Pla
nets_Sun_eph(Markers(2)+2:Markers(3),3))
if Earth == 1
plot3(rel_pos_Planets_Sun(3,1),rel_pos_Planets_Sun(3,2),rel_pos_Planets_Sun(3,3),'b.','MarkerSize',28
)
end
% MARS:
plot3(r_Planets_Sun_eph(Markers(3)+1:Markers(4),1),r_Planets_Sun_eph(Markers(3)+1:Markers(4),2),r_Pla
nets_Sun_eph(Markers(3)+1:Markers(4),3))
if Mars == 1
plot3(rel_pos_Planets_Sun(4,1),rel_pos_Planets_Sun(4,2),rel_pos_Planets_Sun(4,3),'r.','MarkerSize',22
)
end
% SATURN:
plot3(r_Planets_Sun_eph(Markers(5)+1:Markers(6),1),r_Planets_Sun_eph(Markers(5)+1:Markers(6),2),r_Pla
nets_Sun_eph(Markers(5)+1:Markers(6),3))
if Saturn == 1
plot3(rel_pos_Planets_Sun(6,1),rel_pos_Planets_Sun(6,2),rel_pos_Planets_Sun(6,3),'k.','MarkerSize',10
0)
end
% MOON:
if Moon == 1
plot3(rel_pos_Planets_Sun(7,1),rel_pos_Planets_Sun(7,2),rel_pos_Planets_Sun(7,3),'k.','MarkerSize',15
)
end
axis equal
elseif Planet == 6
figure
hold on
% SUN:
plot3(r_Sun(1,1),r_Sun(1,2),r_Sun(1,3),'ok','MarkerFaceColor','y','MarkerSize',10)

% TESTED PLANET:
plot3(r_Planet_Sun_eph(:,1),r_Planet_Sun_eph(:,2),r_Planet_Sun_eph(:,3))
plot3(rel_pos_planet_Sun(1),rel_pos_planet_Sun(2),rel_pos_planet_Sun(3),'.k','MarkerSize',100)
% MERCURY:
plot3(r_Planets_Sun_eph(1:Markers(1),1),r_Planets_Sun_eph(1:Markers(1),2),r_Planets_Sun_eph(1:Markers
(1),3))
if Mercury == 1
plot3(rel_pos_Planets_Sun(1,1),rel_pos_Planets_Sun(1,2),rel_pos_Planets_Sun(1,3),'k.')
end
% VENUS:
plot3(r_Planets_Sun_eph(Markers(1)+1:Markers(2),1),r_Planets_Sun_eph(Markers(1)+1:Markers(2),2),r_Pla
nets_Sun_eph(Markers(1)+1:Markers(2),3))
if Venus == 1
plot3(rel_pos_Planets_Sun(2,1),rel_pos_Planets_Sun(2,2),rel_pos_Planets_Sun(2,3),'m.','MarkerSize',26
)
end
% EARTH:
plot3(r_Planets_Sun_eph(Markers(2)+2:Markers(3),1),r_Planets_Sun_eph(Markers(2)+2:Markers(3),2),r_Pla
nets_Sun_eph(Markers(2)+2:Markers(3),3))
```

```
if Earth == 1
plot3(rel_pos_Planets_Sun(3,1),rel_pos_Planets_Sun(3,2),rel_pos_Planets_Sun(3,3),'b.','MarkerSize',28
)
end
% MARS:
plot3(r_Planets_Sun_eph(Markers(3)+1:Markers(4),1),r_Planets_Sun_eph(Markers(3)+1:Markers(4),2),r_Pla
nets_Sun_eph(Markers(3)+1:Markers(4),3))
if Mars == 1
plot3(rel_pos_Planets_Sun(4,1),rel_pos_Planets_Sun(4,2),rel_pos_Planets_Sun(4,3),'r.','MarkerSize',22
)
end
% JUPITER:
plot3(r_Planets_Sun_eph(Markers(4)+1:Markers(5),1),r_Planets_Sun_eph(Markers(4)+1:Markers(5),2),r_Pla
nets_Sun_eph(Markers(4)+1:Markers(5),3))
if Jupiter == 1
plot3(rel_pos_Planets_Sun(5,1),rel_pos_Planets_Sun(5,2),rel_pos_Planets_Sun(5,3),'.k','MarkerSize',12
0)
end
% MOON:

if Moon == 1
plot3(rel_pos_Planets_Sun(7,1),rel_pos_Planets_Sun(7,2),rel_pos_Planets_Sun(7,3),'k.','MarkerSize',15
)
end
axis equal
end
xlabel('X (km)','FontSize',18)
ylabel('Y (km)','FontSize',18)
zlabel('Z (km)','FontSize',18)
title('Fixed planetary configuration for the random test','FontSize',24)
```

A.5 Validation

Model_BepiColombo_V5.m

```
% --------------------------------------------------------------------------
%%% Propagator model.V5 [definitive] (Be orbiting around the Sun)
% --------------------------------------------------------------------------
tic;
Asteroid = 'BepiColombo'; % It is written like that for not modifying the function "asteroid_cases.m"
Propagation_time = round(date2mjd2000([2025,11,2,8,42,0])-date2mjd2000([2018,10,20,2,13,0]));
Ap = input('Do you want to simulate using 2BP (write 0) or NBP (write 1)?'); % To introduce
perturbations or not
if Ap == 0
   Perturbation_method = 0;
else
   Perturbation_method = input('Do you want to propagate the initial conditions using Cowell (1) or
Encke(2)');
end
if Perturbation_method == 1
   Propagator = input('Do you want to use the static (1) or the dynamic (2) propagator? (Type 1 or
2)');
else
   Propagator = 0;
end
if Ap == 1 % If type kind of NBP is desired
    if Propagator == 1 || Perturbation_method == 2 % When  the static NBP version is chosen, it is
necessary to know which bodies
       disp('Introduce which planets/bodies you want to be considered:\n');
       Mercury = input('Mercury? 1 = YES 0 = NO\n');
       Venus = input('Venus? 1 = YES 0 = NO\n');
       Earth = input('Earth? 1 = YES 0 = NO\n');
       Mars = input('Mars? 1 = YES 0 = NO\n');
       Jupiter = input('Jupiter? 1 = YES 0 = NO\n');
       Saturn = input('Saturn? 1 = YES 0 = NO\n');
       Uranus = input('Uranus? 1 = YES 0 = NO\n');
       Neptune = input('Neptune? 1 = YES 0 = NO\n');
       Pluto = input('Pluto? 1 = YES 0 = NO\n');
       Moon = input('Moon? 1 = YES 0 = NO\n');
    else
       Mercury = 1;Venus = 1;Earth = 1;Mars = 1;Jupiter = 1;Saturn = 1;
       Uranus = 1;Neptune = 1;Pluto = 1;Moon = 1;
    end
else
   Mercury = 0;Venus = 0;Earth = 0;Mars = 0;Jupiter = 0;Saturn = 0;
   Uranus = 0;Neptune = 0;Pluto = 0;Moon = 0;
   Propagator = 1;
end
%% Ancillary data
G = 6.67408e-20;                        % Universal gravitational constant [km^3/(kg.s^2)]
muSun = getAstroConstants('Sun','mu'); % Gravitational parameter for the Sun [km^3/s^2]
M_S = 1988500e24;                       % Sun Mass [kg]
magnification = 20;                     % To properly see the sun in large orbits
R_S = 696000*magnification;             % Sun's radius [km]
% Load of planetary gravitational parameters, masses and radii:
```

```
[mus,Mass,Radii] = Planetary_constants;
%% Load positions and velocities (JPL SSB ephemerides)
% [r_Sun_SSB_eph,v_Sun_SSB_eph,r_Mercury_SSB_eph,v_Mercury_SSB_eph,r_Venus_SSB_eph,v_Venus_SSB_eph...
%
,r_Earth_SSB_eph,v_Earth_SSB_eph,r_Mars_SSB_eph,v_Mars_SSB_eph,r_Jupiter_SSB_eph,v_Jupiter_SSB_eph...
%
,r_Saturn_SSB_eph,v_Saturn_SSB_eph,r_Bepi_SSB_eph,v_Bepi_SSB_eph,r_Uranus_SSB_eph,v_Uranus_SSB_eph...
% ,r_Neptune_SSB_eph,v_Neptune_SSB_eph,r_Pluto_SSB_eph,v_Pluto_SSB_eph,r_Moon_SSB_eph,v_Moon_SSB_eph]
= asteroid_cases(Asteroid,Propagation_time);
%% Numerical propagation (solving the differential equations of motion)
simulation_time = 27000;      % Number of hours simulated of ephemeris trajectory (about 1094 days,
arbitrarily selected only for having those 3 types of flyby)
data_step = 6;                % Data separation time (from HORIZONS ephemeris)
x = zeros(1,6);               % Construction of a vector for initial conditions
x(1:3) = r_Bepi_SSB_eph(1,:); % Initial position of BepiColombo relative to SSB
x(4:6) = v_Bepi_SSB_eph(1,:); % Initial velocity of BepiColombo relative to SSB
r_Bepi_SSB_NBP = zeros(simulation_time/data_step,3);
v_Bepi_SSB_NBP = zeros(simulation_time/data_step,3);
r_Bepi_SSB_NBP(1,:) = x(1:3);
v_Bepi_SSB_NBP(1,:) = x(4:6);
rho = x;                      % In case Encke's method is engaged
dr = [0 0 0 0 0 0];           % In case Encke's method is engaged
step = 1;
stop = 0;
jump = 0;
control = 0;
skip = 0;
rectification = 0; % Ancillary variable for applying Encke's method (if selected)
rho_vector = 0;    % For Encke
k = 1;             % Initialisation (for rectification if it is the case)
epsilon = 0.00015; % Threshold for rectification process (if Encke is chosen)
flyby = 0;         % To obligate Encke's to rectify after a flyby event
% Define vectors to store steps (times) when the S/C is within a given PS
% (for the dynamic integrator, if used):
PS_Records = zeros(1,simulation_time/data_step); % 1: Mercury 2:Venus 3:Earth 4:Mars 5:Jupiter
6:Saturn 7:Moon 13: Overlapping of two or more PSs
options = odeset('RelTol',1e-12,'AbsTol',1e-13);   % specify ODE45 tolerances
tspan = 0:data_step*3600:(simulation_time-1)*3600; % [number of seconds in 2570 days] => Necessary to
setup the solver in s, since it is                                                  % how the
information for initial conditions is get
for i=1:simulation_time/data_step-1
    % Loop to skip all the steps involved on link-conic approximation for
    % flybys:
    if skip > 0 && control < skip
        control = control+1;
        continue
    end
    % Insertion of a block for command detection of SOIs crossing and minimum distances to
    % planets to later numerically simulate each planet flyby (information of distances gattered from
ephemeris):
    % EARTH:
    R_Earth_SOI = norm(r_Earth_SSB_eph(step,:)-r_Sun_SSB_eph(step,:))*(Mass(3)/M_S)^(2/5); % Updated
as the Earth is moving around Sun
    distance_Earth = norm(r_Bepi_SSB_eph(step,:)-r_Earth_SSB_eph(step,:));                 % [km]
    if distance_Earth <= R_Earth_SOI && step >= 60 % Condition for detecting SOI crossing
       initiation = step;
       flag = 'Earth';
       counter = step;
       while norm(r_Bepi_SSB_eph(counter+1,:)-r_Earth_SSB_eph(counter+1,:)) < distance_Earth %
Condition for detecting maximum approach
           counter = counter+1;
           minimum_distance_flag = counter; % Updated each loop until finding the minimum
           distance_Earth = norm(r_Bepi_SSB_eph(counter,:)-r_Earth_SSB_eph(counter,:));
       end
       while distance_Earth <= R_Earth_SOI % Condition for detecting maximum approach
           counter = counter+1;
           finalization = counter;
           R_Earth_SOI = norm(r_Earth_SSB_eph(counter,:)-
r_Sun_SSB_eph(counter,:))*(Mass(3)/M_S)^(2/5); % Updated as the Earth is moving around Sun
           distance_Earth = norm(r_Bepi_SSB_eph(counter,:)-r_Earth_SSB_eph(counter,:));
% [km]
       end
       jump = 1;
    end
    % VENUS:
    R_Venus_SOI = norm(r_Venus_SSB_eph(step,:)-r_Sun_SSB_eph(step,:))*(Mass(2)/M_S)^(2/5); % Updated
as the Earth is moving around Sun
    distance_Venus = norm(r_Bepi_SSB_eph(step,:)-r_Venus_SSB_eph(step,:));                 % [km]
    if distance_Venus <= R_Venus_SOI
       initiation = step; % Updated each loop until finding the minimum
       flag = 'Venus';
       counter = step;
       while norm(r_Bepi_SSB_eph(counter+1,:)-r_Venus_SSB_eph(counter+1,:)) < distance_Venus %
Condition for detecting maximum approach
           counter = counter+1;
           minimum_distance_flag = counter; % Updated each loop until finding the minimum
           distance_Venus = norm(r_Bepi_SSB_eph(counter,:)-r_Venus_SSB_eph(counter,:));
       end
       while distance_Venus <= R_Venus_SOI % Condition for detecting maximum approach
           counter = counter+1;
           finalization = counter;
```

```
            R_Venus_SOI = norm(r_Venus_SSB_eph(counter,:)-
r_Sun_SSB_eph(counter,:))*(Mass(2)/M_S)^(2/5); % Updated as the Earth is moving around Sun
            distance_Venus = norm(r_Bepi_SSB_eph(counter,:)-r_Venus_SSB_eph(counter,:));
% [km]
        end
        jump = 1;
    end
    % MERCURY:
    R_Mercury_SOI = norm(r_Mercury_SSB_eph(step,:)-r_Sun_SSB_eph(step,:))*(Mass(1)/M_S)^(2/5); %
Updated as Venus is moving around Sun
    distance_Mercury = norm(r_Bepi_SSB_eph(step,:)-r_Mercury_SSB_eph(step,:));                  % [km]
    if distance_Mercury <= R_Mercury_SOI
        initiation = step; % Updated each loop until finding the minimum
        flag = 'Mercury';
        counter = step;
        while norm(r_Bepi_SSB_eph(counter+1,:)-r_Mercury_SSB_eph(counter+1,:)) < distance_Mercury %
Condition for detecting maximum approach
            counter = counter+1;
            minimum_distance_flag = counter; % Updated each loop until finding the minimum
            distance_Mercury = norm(r_Bepi_SSB_eph(counter,:)-r_Mercury_SSB_eph(counter,:));
        end
        while distance_Mercury <= R_Mercury_SOI % Condition for detecting maximum approach
            counter = counter+1;
            finalization = counter;
            R_Mercury_SOI = norm(r_Mercury_SSB_eph(counter,:)-
r_Sun_SSB_eph(counter,:))*(Mass(1)/M_S)^(2/5); % Updated as Mercury is moving around Sun
            distance_Mercury = norm(r_Bepi_SSB_eph(counter,:)-r_Mercury_SSB_eph(counter,:));
% [km]
        end
        jump = 1;
    end
%%%%%%%%%%%%%%%%Block for SUN domination (inside Sun's SOI)%%%%%%%%%%%%%%%%%%
    if step >= 60 % During the first 360 h (15 days, the Earth is not considered as perturbation
                  % to avoid Cowell's failures, as the S/C is inside the Earth's SOI and we are not
                  % considering the Earth as the main body)
         stop=1;
    end
    if (distance_Earth >= R_Earth_SOI || step < 60) && jump == 0
        % Its disturbances are:(Common for both type of propagator)
        % MERCURY:
        r32 = x(1:3)-r_Mercury_SSB_eph(i,:);              % BepiColombo's relative position to Mercury
        r31 = r_Sun_SSB_eph(i,:)-r_Mercury_SSB_eph(i,:);  % Sun relative position to Mercury
        % VENUS:
        r42 = x(1:3)-r_Venus_SSB_eph(i,:);                % BepiColombo's relative position to Venus
        r41 = r_Sun_SSB_eph(i,:)-r_Venus_SSB_eph(i,:);    % Sun relative position to Venus
        % EARTH:
        r52 = x(1:3)-r_Earth_SSB_eph(i,:);                % BepiColombo's relative position to Earth
        r51 = r_Sun_SSB_eph(i,:)-r_Earth_SSB_eph(i,:);    % Sun relative position to Earth
        % MARS:
        r62 = x(1:3)-r_Mars_SSB_eph(i,:);                 % BepiColombo's relative position to Mars
        r61 = r_Sun_SSB_eph(i,:)-r_Mars_SSB_eph(i,:);     % Sun relative position to Mars
        % JUPITER:
        r72 = x(1:3)-r_Jupiter_SSB_eph(i,:);              % BepiColombo's relative position to Jupiter
        r71 = r_Sun_SSB_eph(i,:)-r_Jupiter_SSB_eph(i,:);  % Sun relative position to Jupiter
        % SATURN:
        r82 = x(1:3)-r_Saturn_SSB_eph(i,:);               % BepiColombo's relative position to Saturn
        r81 = r_Sun_SSB_eph(i,:)-r_Saturn_SSB_eph(i,:);   % Sun relative position to Saturn
        % URANUS:
        r92 = x(1:3)-r_Uranus_SSB_eph(step,:);            % BepiColombo's relative position to Uranus
        r91 = r_Sun_SSB_eph(i,:)-r_Uranus_SSB_eph(i,:);   % Sun relative position to Uranus
        % NEPTUNE:
        r102 = x(1:3)-r_Neptune_SSB_eph(i,:);             % BepiColombo's relative position to Neptune
        r101 = r_Sun_SSB_eph(i,:)-r_Neptune_SSB_eph(i,:); % Sun relative position to Neptune
        % PLUTO:
        r112 = x(1:3)-r_Pluto_SSB_eph(i,:);               % BepiColombo's relative position to Pluto
        r111 = r_Sun_SSB_eph(i,:)-r_Pluto_SSB_eph(i,:);   % Sun relative position to Pluto
        % MOON:
        r122 = x(1:3)-r_Moon_SSB_eph(i,:);                % BepiColombo's relative position to Moon
        r121 = r_Sun_SSB_eph(i,:)-r_Moon_SSB_eph(i,:);    % Sun relative position to Moon
        % Computation of current magnitude of the resulting perturbative acceleration term:
        A_perturbation_X = -Mercury*Ap*mus(1)*(r32(1)/norm(r32)^3-r31(1)/norm(r31)^3)-
Venus*Ap*mus(2)*(r42(1)/norm(r42)^3-r41(1)/norm(r41)^3)-Earth*stop*Ap*mus(3)*(r52(1)/norm(r52)^3-
r51(1)/norm(r51)^3)-Mars*Ap*mus(4)*(r62(1)/norm(r62)^3-r61(1)/norm(r61)^3)-
Jupiter*Ap*mus(5)*(r72(1)/norm(r72)^3-r71(1)/norm(r71)^3)-Saturn*Ap*mus(6)*(r82(1)/norm(r82)^3-
r81(1)/norm(r81)^3)-Uranus*Ap*mus(7)*(r92(1)/norm(r92)^3-r91(1)/norm(r91)^3)-
Neptune*Ap*mus(8)*(r102(1)/norm(r102)^3-r101(1)/norm(r101)^3)-Pluto*Ap*mus(9)*(r112(1)/norm(r112)^3-
r111(1)/norm(r111)^3)-Moon*Ap*mus(10)*(r122(1)/norm(r122)^3-r121(1)/norm(r121)^3); % X Component
        A_perturbation_Y = -Mercury*Ap*mus(1)*(r32(2)/norm(r32)^3-r31(2)/norm(r31)^3)-
Venus*Ap*mus(2)*(r42(2)/norm(r42)^3-r41(2)/norm(r41)^3)-Earth*stop*Ap*mus(3)*(r52(2)/norm(r52)^3-
r51(2)/norm(r51)^3)-Mars*Ap*mus(4)*(r62(2)/norm(r62)^3-r61(2)/norm(r61)^3)-
Jupiter*Ap*mus(5)*(r72(2)/norm(r72)^3-r71(2)/norm(r71)^3)-Saturn*Ap*mus(6)*(r82(2)/norm(r82)^3-
r81(2)/norm(r81)^3)-Uranus*Ap*mus(7)*(r92(2)/norm(r92)^3-r91(2)/norm(r91)^3)-
Neptune*Ap*mus(8)*(r102(2)/norm(r102)^3-r101(2)/norm(r101)^3)-Pluto*Ap*mus(9)*(r112(2)/norm(r112)^3-
r111(2)/norm(r111)^3)-Moon*Ap*mus(10)*(r122(2)/norm(r122)^3-r121(2)/norm(r121)^3); % Y Component
        A_perturbation_Z = -Mercury*Ap*mus(1)*(r32(3)/norm(r32)^3-r31(3)/norm(r31)^3)-
Venus*Ap*mus(2)*(r42(3)/norm(r42)^3-r41(3)/norm(r41)^3)-Earth*stop*Ap*mus(3)*(r52(3)/norm(r52)^3-
r51(3)/norm(r51)^3)-Mars*Ap*mus(4)*(r62(3)/norm(r62)^3-r61(3)/norm(r61)^3)-
Jupiter*Ap*mus(5)*(r72(3)/norm(r72)^3-r71(3)/norm(r71)^3)-Saturn*Ap*mus(6)*(r82(3)/norm(r82)^3-
r81(3)/norm(r81)^3)-Uranus*Ap*mus(7)*(r92(3)/norm(r92)^3-r91(3)/norm(r91)^3)-
Neptune*Ap*mus(8)*(r102(3)/norm(r102)^3-r101(3)/norm(r101)^3)-Pluto*Ap*mus(9)*(r112(3)/norm(r112)^3-
r111(3)/norm(r111)^3)-Moon*Ap*mus(10)*(r122(3)/norm(r122)^3-r121(3)/norm(r121)^3); % Z Component
```

```
        A_perturbation = sqrt(A_perturbation_X^2+A_perturbation_Y^2+A_perturbation_Z^2);
        % Engages the proper perturbation method (previously selected by the user):
        if Perturbation_method == 2 % Encke's method is applied
        [x,rho_vector,dr,k,epsilon,flyby] = Encke_method_pro(x,i,A_perturbation_X,A_perturbation_Y...
       ,A_perturbation_Z,rho,dr,options,muSun,r_Sun_SSB_eph,v_Sun_SSB_eph,rho_vector,k,epsilon,flyby);
        else % Cowell's method is applied
        switch Propagator
            % Call to the desired numerical propagator:
            case 1 % Static propagator
            [x] = static_propagator(distance_Earth,R_Earth_SOI,step,jump,r_Sun_SSB_eph...
          ,v_Sun_SSB_eph,x,tspan,options,i,muSun,A_perturbation_X,A_perturbation_Y,A_perturbation_Z);
            case 2 % Dynamic propagator
            [x,PS_Records] = dynamic_propagator(step,r_Sun_SSB_eph,v_Sun_SSB_eph,r_Mercury_SSB_eph...
            ,r_Venus_SSB_eph,r_Earth_SSB_eph,r_Mars_SSB_eph,r_Jupiter_SSB_eph,r_Saturn_SSB_eph...
            ,r_Moon_SSB_eph,x,tspan,options,i,Mercury,Venus,Earth,Mars,Jupiter,Saturn,Moon,Ap....
,stop,mus,muSun,A_perturbation,r31,r41,r51,r61,r71,r81,r121,r32,r42,r52,r62,r72,r82,r122,PS_Records);
        end
        end
    end
    if jump == 1
        [r_Bepi_SSB_link_conic,v_Bepi_SSB_link_conic,step,PS_Records] =
flybys(R_Earth_SOI,R_Venus_SOI...
        ,R_Mercury_SOI,data_step,r_Bepi_SSB_eph,v_Bepi_SSB_eph,r_Earth_SSB_eph,v_Earth_SSB_eph...,
        ,initiation,finalization,r_Sun_SSB_eph,r_Venus_SSB_eph,v_Venus_SSB_eph,r_Mercury_SSB_eph...
,v_Mercury_SSB_eph,minimum_distance_flag,flag,mus,options,muSun,PS_Records,Propagator,Perturbation_me
thod);
        % Update position and velocity after the flyby integration event:
        r_Bepi_SSB_NBP(initiation:finalization-1,:) = r_Bepi_SSB_link_conic(1:(finalization-
initiation),:);
        v_Bepi_SSB_NBP(initiation:finalization-1,:) = v_Bepi_SSB_link_conic(1:(finalization-
initiation),:);
        x(1:3) = r_Bepi_SSB_NBP(finalization-1,:);
        x(4:6) = v_Bepi_SSB_NBP(finalization-1,:);
        jump = 0;
        flyby = 1;
        k = 1;
        if strcmp(flag,'Venus') == 1
           skip = skip+finalization-initiation-1;
        elseif strcmp(flag,'Earth') == 1
           skip = finalization-initiation-2;
        else
           skip = skip+finalization-initiation-1;
        end
        step = step+1;
    else
        % Update position and velocity vectors wrt SSB (from NBP propagation):
        r_Bepi_SSB_NBP(i+1,:) = x(end,1:3);
        v_Bepi_SSB_NBP(i+1,:) = x(end,4:6);
        x = x(end,:);
        step = step+1;
    end
end
%% Plotting the final scenario
figure
hold on
% Plot of ephemeris and simulated trajectories:
plot3(r_Bepi_SSB_eph(1:step-1,1),r_Bepi_SSB_eph(1:step-1,2),r_Bepi_SSB_eph(1:step-1,3),'-
b','LineWidth',2)
plot3(r_Bepi_SSB_NBP(1:step-1,1),r_Bepi_SSB_NBP(1:step-1,2),r_Bepi_SSB_NBP(1:step-1,3),'-
r','LineWidth',2.5)
% Plot of planet orbits:
plot3(r_Mercury_SSB_eph(1:step-1,1),r_Mercury_SSB_eph(1:step-1,2),r_Mercury_SSB_eph(1:step-
1,3),'k','LineWidth',1)
plot3(r_Venus_SSB_eph(1:step-1,1),r_Venus_SSB_eph(1:step-1,2),r_Venus_SSB_eph(1:step-
1,3),'k','LineWidth',1)
plot3(r_Earth_SSB_eph(1:step-1,1),r_Earth_SSB_eph(1:step-1,2),r_Earth_SSB_eph(1:step-
1,3),'k','LineWidth',1)
plot3(r_Mars_SSB_eph(1:step-1,1),r_Mars_SSB_eph(1:step-1,2),r_Mars_SSB_eph(1:step-
1,3),'k','LineWidth',1)
% Plot of planet positions in real time (for simbolical representation purposes only):
[X,Y,Z] = sphere;
Sun = surf(X*R_S+r_Sun_SSB_eph(step-1,1),Y*R_S+r_Sun_SSB_eph(step-1,2),Z*R_S+r_Sun_SSB_eph(step-
1,3),'edgealpha',0);
set(Sun, 'FaceColor', [1 1 0])
[X,Y,Z] = sphere;
Mercury = surf(X*Radii(1)+r_Mercury_SSB_eph(step-1,1),Y*Radii(1)+r_Mercury_SSB_eph(step-
1,2),Z*Radii(1)+r_Mercury_SSB_eph(step-1,3),'edgealpha',0);
set(Mercury,'FaceColor', [0 0 0])
[X,Y,Z] = sphere;
Venus = surf(X*Radii(2)+r_Venus_SSB_eph(step-1,1),Y*Radii(2)+r_Venus_SSB_eph(step-
1,2),Z*Radii(2)+r_Venus_SSB_eph(step-1,3),'edgealpha',0);
set(Venus, 'FaceColor', [0.4 0.4 0.4])
[X,Y,Z] = sphere;
Earth = surf(X*Radii(3)+r_Earth_SSB_eph(step-1,1),Y*Radii(3)+r_Earth_SSB_eph(step-
1,2),Z*Radii(3)+r_Earth_SSB_eph(step-1,3),'edgealpha',0);
set(Earth, 'FaceColor', 'b')
[X,Y,Z] = sphere;
Mars = surf(X*Radii(4)+r_Mars_SSB_eph(step-1,1),Y*Radii(4)+r_Mars_SSB_eph(step-
1,2),Z*Radii(4)+r_Mars_SSB_eph(step-1,3),'edgealpha',0);
set(Mars, 'FaceColor', 'r')
axis equal
```

```
xlabel('X (km)','FontSize',18)
ylabel('Y (km)','FontSize',18)
zlabel('Z (km)','FontSize',18)
title('BepiColombo trajectories from Oct 21th of 2018 to Nov 2nd of 2025 (Final scenario at
arrival)','FontSize',18)
legend('\fontsize{16} High-precision ephemerides for BepiColombo trajectory','\fontsize{16}
BepiColombo trajectory by NBP model','\fontsize{16} High-precision ephemerides for Mercury
trajectory','\fontsize{16} High-precision ephemerides for Venus trajectory','\fontsize{16} High-
precision ephemerides for Earth trajectory','\fontsize{16} High-precision ephemerides for Mars
trajectory','\fontsize{16} Sun (20 times magnified)','\fontsize{16} Mercury (1000 times
magnified)','\fontsize{16} Venus (1000 times magnified)','\fontsize{16} Earth (1000 times
magnified)','\fontsize{16} Mars (1000 times magnified)')
%% Evaluating the discrepances
% figure
r_error = zeros(1,length(r_Bepi_SSB_NBP));
times = 0:data_step*3600:(simulation_time-1)*3600;
for i=1:length(r_Bepi_SSB_NBP)
r_error(1,i) = norm(r_Bepi_SSB_NBP(i,:)-r_Bepi_SSB_eph(i,:));
end
plot(times,r_error,'LineWidth',1.5)
xlabel('\fontsize{16}Time (s)')
ylabel('\fontsize{16}Error in position (km)')
title('\fontsize{18}Discrepancy between NBP computed trajectory and ephemeris JPL')
%%  PS crossover plot (to check within which Perturbation Spheres the S/C enters during simulation)
% Keycode:
% 1:Mercury 2:Venus 3:Earth 4:Mars 5:Jupiter 6:Saturn 7:Moon
% 13: A coded number to indicate the overlap of two or more PS (S/C within two or more PS
simultaneously)
if Propagator == 2 % In case dynamic propagator is engaged, it plots a record of planet PSs where the
S/C entered
   plot(times,PS_Records)
   ylim([0 7])
   title('\fontsize{20} Perturbation Sphere crossover plot versus time')
   xlabel('\fontsize{18} Time (s)')
   ylabel('\fontsize{19} Number')
   legend('\fontsize{16} 1: Mercury 2: Venus 3: Earth 4: Mars 5: Jupiter 6: Saturn 7: Moon 13: A
coded number to indicate the overlap of two or more PS')
end
toc
%% State changes over simulation time (Manipulate apart as a section)
clearvars -except simulation_time data_step; % Since one execution only gives one case (comment if
% a specific variable is desired to be conserved)
% (Uncomment those desired ones)
load test_states_2BP
load test_states_NBP
% load test_states_NBP_partial
% load test_states_NBP_partial_dynamic
load Ephemeris_data_6h
times = 0:data_step*3600:(simulation_time-1)*3600; % [s]
% Using subplot, plot each position and velocity over time:
figure; % Position
subplot(3,1,1);
plot(times,r_Bepi_SSB_NBP(:,1),'b',times,r_Bepi_SSB_2BP(:,1),'r');
title('\fontsize{20}States Over Simulation Time');
ylabel('X (km)')
legend('\fontsize{16} Position with NBP','\fontsize{16} Position with 2BP');
grid on;
subplot(3,1,2);
plot(times,r_Bepi_SSB_NBP(:,2),'b',times,r_Bepi_SSB_2BP(:,2),'r');
ylabel('Y (km)')
legend('\fontsize{16} Position with NBP','\fontsize{16} Position with 2BP');
grid on;
subplot(3,1,3);
plot(times,r_Bepi_SSB_NBP(:,3),'b',times,r_Bepi_SSB_2BP(:,3),'r');
ylabel('Z (km)')
xlabel('Time (h)')
legend('\fontsize{16} Position with NBP','\fontsize{16} Position with 2BP');
grid on;
figure; % Velocities
subplot(3,1,1);
plot(times,v_Bepi_SSB_NBP(:,1),'b',times,v_Bepi_SSB_2BP(:,1),'r');
title('\fontsize{20}States Over Simulation Time');
ylabel('X (km/s)')
legend('\fontsize{16} Velocity with NBP','\fontsize{16} Velocity with 2BP');
grid on;
subplot(3,1,2);
plot(times,v_Bepi_SSB_NBP(:,2),'b',times,v_Bepi_SSB_2BP(:,2),'r');
ylabel('Y (km/s)')
legend('\fontsize{16} Velocity with NBP','\fontsize{16} Velocity with 2BP');
grid on;
subplot(3,1,3);
plot(times,v_Bepi_SSB_NBP(:,3),'b',times,v_Bepi_SSB_2BP(:,3),'r');
ylabel('Z (km/s)')
xlabel('Time (h)')
legend('\fontsize{16} Velocity with NBP','\fontsize{16} Velocity with 2BP');
grid on;
%% Analyses made for ephemeris trajectory
% Computing magnitude of the Sun's perturbed gravitational field for Asteroid
% instantaneous position:
gravitational_fields_BepiColombo(G,r_Bepi_SSB_eph,r_Sun_SSB_eph,r_Mercury_SSB_eph...
,r_Venus_SSB_eph,r_Earth_SSB_eph,r_Mars_SSB_eph,r_Jupiter_SSB_eph,r_Saturn_SSB_eph...
```

```
,Mass,Propagation_time)
% Computing the instantaneous distances from BepiColombo to each celestial body:
[time_within_PZ,day_start,day_finish,path_sections,times] =
distances_BepiColombo(r_Mercury_SSB_eph,r_Venus_SSB_eph,r_Earth_SSB_eph,r_Mars_SSB_eph...
,r_Jupiter_SSB_eph,r_Saturn_SSB_eph,Propagation_time,r_Bepi_SSB_eph);
% Animated trajectories: (Better for 1day time step of ephemeris data)
Animation = input('Do you want to visualize an animation? (1 = YES, 0 = NO)');
Animation_plot_BepiColombo(Animation,r_Sun_SSB_eph,r_Mercury_SSB_eph,r_Venus_SSB_eph...
,r_Earth_SSB_eph,r_Mars_SSB_eph,r_Bepi_SSB_eph,R_S,TP_Mercury,TP_Venus,TP_Earth,TP_Mars)
```

flybys.m

```
function [r_Bepi_SSB_link_conic,v_Bepi_SSB_link_conic,step,PS_Records] =
flybys(R_Earth_SOI,R_Venus_SOI...
    ,R_Mercury_SOI,data_step,r_Bepi_SSB_eph,v_Bepi_SSB_eph,r_Earth_SSB_eph,v_Earth_SSB_eph...,
    ,initiation,finalization,r_Sun_SSB_eph,r_Venus_SSB_eph,v_Venus_SSB_eph,r_Mercury_SSB_eph...
    ,v_Mercury_SSB_eph,minimum_distance_flag,flag,mus,options,muSun,PS_Records,Propagator...
    ,Perturbation_method)
switch flag
    case 'Earth'
    % States related to entry to the SOI and at the pericenter of the hyperbola:
    vi = v_Bepi_SSB_eph(initiation,:)-v_Earth_SSB_eph(initiation,:); % Vector velocity at SOI entry
relative to Earth
    % The semi-major axis and the eccentricity of the hyperbolic conic can
    % be easily obtained from the incoming and from the pericenter radius:
    a = -mus(3)/norm(vi)^2;
    rp = r_Bepi_SSB_eph(minimum_distance_flag,:)-r_Earth_SSB_eph(minimum_distance_flag,:); % Position
vector at pericenter radius relative to Earth
    e = 1 - norm(rp)/a;
    % Calculate the initial true anomaly:
    theta = acos((a*(1-e^2)/R_Earth_SOI-1)/e);
    case 'Venus'
    % States related to entry to the SOI and at the pericenter of the hyperbola:
    vi = v_Bepi_SSB_eph(initiation,:)-v_Venus_SSB_eph(initiation,:); % Vector velocity at SOI entry
relative to Venus
    % The semi-major axis and the eccentricity of the hyperbolic conic can
    % be easily obtained from the incoming and from the pericenter radius:
    a = -mus(3)/norm(vi)^2;
    rp = r_Bepi_SSB_eph(minimum_distance_flag,:)-r_Venus_SSB_eph(minimum_distance_flag,:); % Position
vector at pericenter radius relative to Venus
    e = 1 - norm(rp)/a;
    % Calculate the initial true anomaly:
    theta = acos((a*(1-e^2)/R_Venus_SOI-1)/e);
     case 'Mercury'
    % States related to entry to the SOI and at the pericenter of the hyperbola:
    vi = v_Bepi_SSB_eph(initiation,:)-v_Mercury_SSB_eph(initiation,:); % Vector velocity at SOI entry
relative to Mercury
    % The semi-major axis and the eccentricity of the hyperbolic conic can
    % be easily obtained from the incoming and from the pericenter radius:
    a = -mus(3)/norm(vi)^2;
    rp = r_Bepi_SSB_eph(minimum_distance_flag,:)-r_Mercury_SSB_eph(minimum_distance_flag,:); %
Position vector at pericenter radius relative to Mercury
    e = 1 - norm(rp)/a;
    % Calculate the initial true anomaly:
    theta = acos((a*(1-e^2)/R_Mercury_SOI-1)/e);
end
%% Common intermediate calculations for period of the hyperbolic flyby manoeuvre
% Then, evaluate the hyperbolic anomaly:
F = 2*atanh(sqrt((e-1)/(e+1))*tan(theta/2));
% Also the initial mean anomaly:
M0 = e*sinh(F)-F;
% Finally, the period of the flyby manoeuvre (the time spent within the
% SOI) is calculated as:
T = 2*M0*sqrt(-a^3/mus(3));
%% Common variables (previous to propagation):
time_step = data_step*3600;                    % Time in seconds of separation for ephemeris data (for
propagation part)
rp = r_Bepi_SSB_eph(minimum_distance_flag,:); % Position vector at pericenter radius relative to SSB
vp = v_Bepi_SSB_eph(minimum_distance_flag,:);            % Velocity vector at pericenter radius
relative to SSB
%% Integration backwards in time (for matching SOI's entry)
tspan = T/2:-time_step:0;
if length(tspan) < 2
    tspan = [tspan 0];
end
r_Bepi_SSB_link_conic = zeros(2*length(tspan),3); % Velocity vector relative to SSB (valid for both
integrations)
v_Bepi_SSB_link_conic = zeros(2*length(tspan),3); % Velocity vector relative to SSB (valid for both
integrations)
x = [rp vp];                                       % State vector containing initial conditions
(pericenter of hyperbola)
r_Bepi_SSB_link_conic(length(tspan),:) = x(1:3);
v_Bepi_SSB_link_conic(length(tspan),:) = x(4:6);
counter = 1;
step = minimum_distance_flag;
for i=minimum_distance_flag-1:-1:initiation
    if strcmp(flag,'Earth') == 1
       % SUN: (Bepi already was initialised wr the Sun)
       r52 = x(1:3);                                          % BepiColombo's relative position to
Sun
```

```
        r51 = r_Earth_SSB_eph(step,:)-r_Sun_SSB_eph(step,:);   % Earth relative position to Sun
        x(1:3) = x(1:3)-r_Earth_SSB_eph(step,:);
        x(4:6) = x(4:6)-v_Earth_SSB_eph(step,:);
        [~,x] = ode45(@(t,x)[x(4);x(5);x(6);...
        -mus(3)*x(1)/norm(x(1:3))^3-muSun*(r52(1)/norm(r52)^3-r51(1)/norm(r51)^3);... % X Component
        -mus(3)*x(2)/norm(x(1:3))^3-muSun*(r52(2)/norm(r52)^3-r51(2)/norm(r51)^3);... % Y Component
        -mus(3)*x(3)/norm(x(1:3))^3-muSun*(r52(3)/norm(r52)^3-r51(3)/norm(r51)^3)]... % Z Component
        ,tspan(counter:counter+1),x,options);
        x(end,1:3) = x(end,1:3)+r_Earth_SSB_eph(step-1,:);
        x(end,4:6) = x(end,4:6)+v_Earth_SSB_eph(step-1,:);
        % Block for accounting the step time within the PS of the flyby
        % planet (only for dynamic propagation):
        if Propagator == 2
           PS_Records(i) = 3;
        end
    elseif strcmp(flag,'Venus') == 1
        % SUN: (Bepi already was initialised wr the Sun)
        r52 = x(1:3);                                          % BepiColombo's relative position to
Sun
        r51 = r_Venus_SSB_eph(step,:)-r_Sun_SSB_eph(step,:);   % Venus relative position to Sun
        x(1:3) = x(1:3)-r_Venus_SSB_eph(step,:);
        x(4:6) = x(4:6)-v_Venus_SSB_eph(step,:);
        [~,x] = ode45(@(t,x)[x(4);x(5);x(6);...
        -mus(2)*x(1)/norm(x(1:3))^3-muSun*(r52(1)/norm(r52)^3-r51(1)/norm(r51)^3);... % X Component
        -mus(2)*x(2)/norm(x(1:3))^3-muSun*(r52(2)/norm(r52)^3-r51(2)/norm(r51)^3);... % Y Component
        -mus(2)*x(3)/norm(x(1:3))^3-muSun*(r52(3)/norm(r52)^3-r51(3)/norm(r51)^3)]... % Z Component
        ,tspan(counter:counter+1),x,options);
        x(end,1:3) = x(end,1:3)+r_Venus_SSB_eph(step-1,:);
        x(end,4:6) = x(end,4:6)+v_Venus_SSB_eph(step-1,:);
        % Block for accounting the step time within the PS of the flyby
        % planet (only for dynamic propagation):
        if Propagator == 2
           PS_Records(i) = 2;
        end
    elseif strcmp(flag,'Mercury') == 1
        % SUN: (Bepi already was initialised wr the Sun)
        r52 = x(1:3);                                          % BepiColombo's relative position to
Sun
        r51 = r_Mercury_SSB_eph(step,:)-r_Sun_SSB_eph(step,:); % Mercury relative position to Sun
        x(1:3) = x(1:3)-r_Mercury_SSB_eph(step,:);
        x(4:6) = x(4:6)-v_Mercury_SSB_eph(step,:);
        [~,x] = ode45(@(t,x)[x(4);x(5);x(6);...
        -mus(1)*x(1)/norm(x(1:3))^3-muSun*(r52(1)/norm(r52)^3-r51(1)/norm(r51)^3);... % X Component
        -mus(1)*x(2)/norm(x(1:3))^3-muSun*(r52(2)/norm(r52)^3-r51(2)/norm(r51)^3);... % Y Component
        -mus(1)*x(3)/norm(x(1:3))^3-muSun*(r52(3)/norm(r52)^3-r51(3)/norm(r51)^3)]... % Z Component
        ,tspan(counter:counter+1),x,options);
        x(end,1:3) = x(end,1:3)+r_Mercury_SSB_eph(step-1,:);
        x(end,4:6) = x(end,4:6)+v_Mercury_SSB_eph(step-1,:);
        % Block for accounting the step time within the PS of the flyby
        % planet (only for dynamic propagation):
        if Propagator == 2
           PS_Records(i) = 1;
        end
    end
    r_Bepi_SSB_link_conic(length(tspan)-counter,:) = x(end,1:3);
    v_Bepi_SSB_link_conic(length(tspan)-counter,:) = x(end,4:6);
    x = x(end,:);
    counter = counter+1;
    step = step-1;
end
%% Integration forwards in time (for matching SOI's exit)
x = [rp vp]; % State vector containing initial conditions (pericenter of hyperbola)
tspan = 0:time_step:length(tspan)*time_step;
if length(tspan) < 2
    tspan = [tspan 0];
end
counter = 1;
step = minimum_distance_flag;
for i=minimum_distance_flag+1:finalization-1
    if strcmp(flag,'Earth') == 1
        % SUN: (Bepi already was initialised wr the Sun)

        r52 = x(1:3);                                          % BepiColombo's relative position to
Sun
        r51 = r_Earth_SSB_eph(step,:)-r_Sun_SSB_eph(step,:);   % Earth relative position to Sun
        x(1:3) = x(1:3)-r_Earth_SSB_eph(step,:);
        x(4:6) = x(4:6)-v_Earth_SSB_eph(step,:);

        [~,x] = ode45(@(t,x)[x(4);x(5);x(6);...
        -mus(3)*x(1)/norm(x(1:3))^3-muSun*(r52(1)/norm(r52)^3-r51(1)/norm(r51)^3);... % X Component
        -mus(3)*x(2)/norm(x(1:3))^3-muSun*(r52(2)/norm(r52)^3-r51(2)/norm(r51)^3);... % Y Component
        -mus(3)*x(3)/norm(x(1:3))^3-muSun*(r52(3)/norm(r52)^3-r51(3)/norm(r51)^3)]... % Z Component
        ,tspan(counter:counter+1),x,options);
        x(end,1:3) = x(end,1:3)+r_Earth_SSB_eph(step+1,:);
        x(end,4:6) = x(end,4:6)+v_Earth_SSB_eph(step+1,:);
        % Block for accounting the step time within the PS of the flyby
        % planet (only for dynamic propagation):
        if Propagator == 2
           PS_Records(i) = 3;
        end
    elseif strcmp(flag,'Venus') == 1
```

```
        % SUN: (Bepi already was initialised wr the Sun)

       r52 = x(1:3);                                          % BepiColombo's relative position to
Sun
       r51 = r_Venus_SSB_eph(step,:)-r_Sun_SSB_eph(step,:);   % Venus relative position to Sun
       x(1:3) = x(1:3)-r_Venus_SSB_eph(step,:);
       x(4:6) = x(4:6)-v_Venus_SSB_eph(step,:);
       [~,x] = ode45(@(t,x)[x(4);x(5);x(6);...
       -mus(2)*x(1)/norm(x(1:3))^3-muSun*(r52(1)/norm(r52)^3-r51(1)/norm(r51)^3);... % X Component
       -mus(2)*x(2)/norm(x(1:3))^3-muSun*(r52(2)/norm(r52)^3-r51(2)/norm(r51)^3);... % Y Component
       -mus(2)*x(3)/norm(x(1:3))^3-muSun*(r52(3)/norm(r52)^3-r51(3)/norm(r51)^3)]... % Z Component
       ,tspan(counter:counter+1),x,options);
       x(end,1:3) = x(end,1:3)+r_Venus_SSB_eph(step+1,:);
       x(end,4:6) = x(end,4:6)+v_Venus_SSB_eph(step+1,:);
       % Block for accounting the step time within the PS of the flyby
       % planet (only for dynamic propagation):
       if Propagator == 2
          PS_Records(i) = 2;
       end
    elseif strcmp(flag,'Mercury') == 1
       % SUN: (Bepi already was initialised wr the Sun)
       r52 = x(1:3);                                          % BepiColombo's relative position to
Sun
       r51 = r_Mercury_SSB_eph(step,:)-r_Sun_SSB_eph(step,:);   % Mercury relative position to Sun
       x(1:3) = x(1:3)-r_Mercury_SSB_eph(step,:);
       x(4:6) = x(4:6)-v_Mercury_SSB_eph(step,:);
       [~,x] = ode45(@(t,x)[x(4);x(5);x(6);...
       -mus(1)*x(1)/norm(x(1:3))^3-muSun*(r52(1)/norm(r52)^3-r51(1)/norm(r51)^3);... % X Component
       -mus(1)*x(2)/norm(x(1:3))^3-muSun*(r52(2)/norm(r52)^3-r51(2)/norm(r51)^3);... % Y Component
       -mus(1)*x(3)/norm(x(1:3))^3-muSun*(r52(3)/norm(r52)^3-r51(3)/norm(r51)^3)]... % Z Component
       ,tspan(counter:counter+1),x,options);
       x(end,1:3) = x(end,1:3)+r_Mercury_SSB_eph(step+1,:);
       x(end,4:6) = x(end,4:6)+v_Mercury_SSB_eph(step+1,:);
       % Block for accounting the step time within the PS of the flyby
       % planet (only for dynamic propagation):
       if Propagator == 2
          PS_Records(i) = 1;
       end
    end
    r_Bepi_SSB_link_conic(length(r_Bepi_SSB_link_conic)/2+counter,:) = x(end,1:3);
    v_Bepi_SSB_link_conic(length(r_Bepi_SSB_link_conic)/2+counter,:) = x(end,4:6);
    x = x(end,:);
    counter = counter+1;
    step = step+1;
end
%% Final loop for detection of zeros in the matrices (to remove them)
if Perturbation_method == 1
   k=0;
   for i=1:length(r_Bepi_SSB_link_conic)
       if k < 2 % Given that the length of the vector becomes shorthened
          if r_Bepi_SSB_link_conic(i,:) == 0
             if strcmp(flag,'Mercury') == 1
                continue;
             end
           r_Bepi_SSB_link_conic(i,:) = [];
           v_Bepi_SSB_link_conic(i,:) = [];
           k = k+1;
          end
       end
   end
else
    j=1;
       while r_Bepi_SSB_link_conic(j,:) == 0 % Given that the length of the vector becomes shorthened
           r_Bepi_SSB_link_conic(j,:) = [];
           v_Bepi_SSB_link_conic(j,:) = [];
       end
end
```

static_propagator.m

```
function [x] = static_propagator(distance_Earth,R_Earth_SOI,step,jump,r_Sun_SSB_eph...
         ,v_Sun_SSB_eph,x,tspan,options,i,muSun,A_perturbation_X,A_perturbation_Y...
         ,A_perturbation_Z)

if (distance_Earth >= R_Earth_SOI || step < 60) && jump == 0
    % Transformation of position and velocity vectors from SSB to relative to Sun:
    x(1:3) = x(1:3)-r_Sun_SSB_eph(i,:);
    x(4:6) = x(4:6)-v_Sun_SSB_eph(i,:);
    % Numerical integration (propagation of trajectory):
    [t,x] = ode45(@(t,x)[x(4);x(5);x(6);...
     -muSun*x(1)/norm(x(1:3))^3+A_perturbation_X;... % X Component
     -muSun*x(2)/norm(x(1:3))^3+A_perturbation_Y;... % Y Component
     -muSun*x(3)/norm(x(1:3))^3+A_perturbation_Z]... % Z Component
     ,tspan(i:i+1),x,options);
     % Transformation of position and velocity vectors to SSB again:
     x(end,1:3) = x(end,1:3)+r_Sun_SSB_eph(i+1,:);
```

```
     x(end,4:6) = x(end,4:6)+v_Sun_SSB_eph(i+1,:);
end
end
```

dynamic_propagator.m

```
function [x,PS_Records] =
dynamic_propagator(step,r_Sun_SSB_eph,v_Sun_SSB_eph,r_Mercury_SSB_eph...
,r_Venus_SSB_eph,r_Earth_SSB_eph,r_Mars_SSB_eph,r_Jupiter_SSB_eph,r_Saturn_SSB_eph,r_Moon_SSB_eph...
,x,tspan,options,i,Mercury,Venus,Earth,Mars,Jupiter,Saturn,Moon,Ap,stop,mus,muSun,A_perturbation...
,r31,r41,r51,r61,r71,r81,r121,r32,r42,r52,r62,r72,r82,r122,PS_Records)

% Ancillary data:
intersection = 0;             % Control variable to check if the S/C is within two or more PS at the
same time
% Index to establish the bodies to be included over the time:
Mercury_IN = 0;Venus_IN = 0;Earth_IN = 0;Mars_IN = 0;Jupiter_IN = 0;
Saturn_IN = 0;Moon_IN = 0;
% Define the limits of each planetary Perturbation Sphere:
thresholds = [6.11e6 2.25e7 2.48e7 8.87e6 5.06e8 2.69e8 2.75e6]; % Mercury, Venus, Earth, Mars,
Jupiter, Saturn & Moon (in this order)
% Define the threshold for the perturbing term magnitude (based on
% BepiColombo's plant propulsive maximum capacity):
threshold = 6.94e-10; % [km/s2]
% Transformation of position and velocity vectors from SSB to relative to Sun:
x(1:3) = x(1:3)-r_Sun_SSB_eph(i,:);
x(4:6) = x(4:6)-v_Sun_SSB_eph(i,:);
if A_perturbation >= threshold && step >=60 % Detection of PS penetration
   % Block of conditional sentences to identify within which PS % is the S/C at the current step:
   if norm(x(1:3)-r_Mercury_SSB_eph(step,:)) <= thresholds(1)   % Check Mercury PS
      Mercury_IN = 1;
      PS_Records(i) = 1;
      intersection = intersection+1;
   end
   if norm(x(1:3)-r_Venus_SSB_eph(step,:)) <= thresholds(2)   % Check Venus PS
      Venus_IN = 1;
      PS_Records(i) = 2;
      intersection = intersection+1;
   end
   if norm(x(1:3)-r_Earth_SSB_eph(step,:)) <= thresholds(3)   % Check Earth PS
      Earth_IN = 1;
      PS_Records(i) = 3;
      intersection = intersection+1;
   end
   if norm(x(1:3)-r_Mars_SSB_eph(step,:)) <= thresholds(4)    % Check Mars PS
      Mars_IN = 1;
      PS_Records(i) = 4;
      intersection = intersection+1;
   end
   if norm(x(1:3)-r_Jupiter_SSB_eph(step,:)) <= thresholds(5) % Check Jupiter PS
      Jupiter_IN = 1;
      PS_Records(i) = 5;
      intersection = intersection+1;
   end
   if norm(x(1:3)-r_Saturn_SSB_eph(step,:)) <= thresholds(6)  % Check Saturn PS
      Saturn_IN = 1;
      PS_Records(i) = 6;
      intersection = intersection+1;
   end
%    if norm(x(1:3)-r_Moon_SSB_eph(step,:)) <= thresholds(7)    % Check Moon PS
%       Moon_IN = 1;
%       PS_Records(i) = 7;
%       intersection = intersection+1;
%    end

   % Additional conditional sentence to identify a hypothetical instant when the S/C
   % could be within two or more PS at the same time:
   if intersection >= 2
      PS_Records(i) = 13; % A codified number to indicate overlapping of PSs
   end
else % Condition to return the control to 2BP propagation when the S/C is above the threshold
     Mercury_IN = 0;Venus_IN = 0;Earth_IN = 0;Mars_IN = 0;Jupiter_IN = 0;
     Saturn_IN = 0;Moon_IN = 0;
end
   % NOTE: - For BepiColombo's case, it must never enter into >= Mars conditions (just Mercury, Venus
and Earth)
% Numerical integration (propagation of trajectory):
[t,x] = ode45(@(t,x)[x(4);x(5);x(6);...
-muSun*x(1)/norm(x(1:3))^3-Mercury*Mercury_IN*Ap*mus(1)*(r32(1)/norm(r32)^3-r31(1)/norm(r31)^3)-
Venus*Venus_IN*Ap*mus(2)*(r42(1)/norm(r42)^3-r41(1)/norm(r41)^3)-
Earth*Earth_IN*stop*Ap*mus(3)*(r52(1)/norm(r52)^3-r51(1)/norm(r51)^3)-
Mars*Mars_IN*Ap*mus(4)*(r62(1)/norm(r62)^3-r61(1)/norm(r61)^3)-
Jupiter*Jupiter_IN*Ap*mus(5)*(r72(1)/norm(r72)^3-r71(1)/norm(r71)^3)-
Saturn*Saturn_IN*Ap*mus(6)*(r82(1)/norm(r82)^3-r81(1)/norm(r81)^3)-
Moon*Moon_IN*Ap*mus(10)*(r122(1)/norm(r122)^3-r121(1)/norm(r121)^3);... % X Component
-muSun*x(2)/norm(x(1:3))^3-Mercury*Mercury_IN*Ap*mus(1)*(r32(2)/norm(r32)^3-r31(2)/norm(r31)^3)-
Venus*Venus_IN*Ap*mus(2)*(r42(2)/norm(r42)^3-r41(2)/norm(r41)^3)-
Earth*Earth_IN*stop*Ap*mus(3)*(r52(2)/norm(r52)^3-r51(2)/norm(r51)^3)-
Mars*Mars_IN*Ap*mus(4)*(r62(2)/norm(r62)^3-r61(2)/norm(r61)^3)-
```

```
Jupiter*Jupiter_IN*Ap*mus(5)*(r72(2)/norm(r72)^3-r71(2)/norm(r71)^3)-
Saturn*Saturn_IN*Ap*mus(6)*(r82(2)/norm(r82)^3-r81(2)/norm(r81)^3)-
Moon*Moon_IN*Ap*mus(10)*(r122(2)/norm(r122)^3-r121(2)/norm(r121)^3);... % Y Component
-muSun*x(3)/norm(x(1:3))^3-Mercury*Mercury_IN*Ap*mus(1)*(r32(3)/norm(r32)^3-r31(3)/norm(r31)^3)-
Venus*Venus_IN*Ap*mus(2)*(r42(3)/norm(r42)^3-r41(3)/norm(r41)^3)-
Earth*Earth_IN*stop*Ap*mus(3)*(r52(3)/norm(r52)^3-r51(3)/norm(r51)^3)-
Mars*Mars_IN*Ap*mus(4)*(r62(3)/norm(r62)^3-r61(3)/norm(r61)^3)-
Jupiter*Jupiter_IN*Ap*mus(5)*(r72(3)/norm(r72)^3-r71(3)/norm(r71)^3)-
Saturn*Saturn_IN*Ap*mus(6)*(r82(3)/norm(r82)^3-r81(3)/norm(r81)^3)-
Moon*Moon_IN*Ap*mus(10)*(r122(3)/norm(r122)^3-r121(3)/norm(r121)^3)]... % Z Component
,tspan(i:i+1),x,options);
% Transformation of position and velocity vectors to SSB again:
x(end,1:3) = x(end,1:3)+r_Sun_SSB_eph(i+1,:);
x(end,4:6) = x(end,4:6)+v_Sun_SSB_eph(i+1,:);
end
```

Planetary_constants.m

```
function [mus,Mass,Radii] = Planetary_constants

%% Planet gravitational parameters

muMer = getAstroConstants('Mercury','mu'); % Gravitational parameter for Mercury
[km^3/s^2]
muVen = getAstroConstants('Venus','mu');   % Gravitational parameter for Venus
[km^3/s^2]
muEarth = getAstroConstants('Earth','mu'); % Gravitational parameter for the Earth
[km^3/s^2]
muMars = getAstroConstants('Mars','mu');   % Gravitational parameter for Mars [km^3/s^2]
muJup = getAstroConstants('Jupiter','mu'); % Gravitational parameter for Jupiter
[km^3/s^2]
muSat = getAstroConstants('Saturn','mu');  % Gravitational parameter for Saturn
[km^3/s^2]
muUr = getAstroConstants('Uranus','mu');   % Gravitational parameter for Uranus
[km^3/s^2]
muNep = getAstroConstants('Neptune','mu'); % Gravitational parameter for Neptune
[km^3/s^2]
muPluto = 869.33907803;                    % Gravitational parameter for Saturn
[km^3/s^2]
muMoon = getAstroConstants('Moon','mu');   % Gravitational parameter for Moon [km^3/s^2]
mus = [muMer muVen muEarth muMars muJup muSat muUr muNep muPluto muMoon];
%% Planet masses
M_Mercury = 3.302e23;   % Mercury's Mass [kg]
M_Venus = 48.685e23;    % Venus Mass [kg]
M_Earth = 5.97219e24;   % Earth's Mass [kg]
M_Mars = 6.4171e23;     % Mars Mass [kg]
M_Jupiter = 1898.13e24; % Jupiter Mass [kg]
M_Saturn = 5.6834e26;   % Saturn Mass [kg]
M_Uranus = 86.813e24;   % Uranus Mass [kg]
M_Neptune = 102.413e24; % Neptune Mass [kg]
M_Pluto = 1.307e22;     % Pluto Mass [kg]
M_Moon = 7.349e22;      % Moon Mass [kg]
Mass = [M_Mercury M_Venus M_Earth M_Mars M_Jupiter M_Saturn M_Uranus M_Neptune M_Pluto
M_Moon];

%% Planet radii (only for simbolical representation purposes)

R_Mer = 2440*1000;      % Mercury's radius (1000 times magnified) [km]
R_Ven = 6051.84*1000;   % Venus radius (1000 times magnified) [km]
R_Earth = 6371.01*1000; % Earth radius (1000 times magnified) [km]
R_Mars = 3389.92*1000;  % Mars radius (1000 times magnified) [km]
Radii = [R_Mer R_Ven R_Earth R_Mars];
```

Animation_plot_BepiColombo.m

```
function Animation_plot_BepiColombo(Animation,r_Sun_SSB_eph,r_Mercury_SSB_eph,r_Venus_SSB_eph...
    ,r_Earth_SSB_eph,r_Mars_SSB_eph,r_Bepi_SSB_eph,R_S,TP_Mercury,TP_Venus,TP_Earth,TP_Mars)

if Animation ==1
index_Mer = 0;
index_Ven = 0;
index_Earth = 0;
index_Mars = 0;
% index_Jup = 0;
% index_Sat = 0;
j_Mer = 1;
j_Ven = 1;
j_Earth = 1;
j_Mars = 1;
% j_Jup = 1;
% j_Sat = 1;
```

```
k_Mer = 1;
k_Ven = 1;
k_Earth = 1;
k_Mars = 1;
% k_Jup = 1;
% k_Sat = 1;
aux_Mer = 1;
aux_Ven = 1;
aux_Earth = 1;
aux_Mars = 1;
% aux_Jup = 1;
% aux_Sat = 1;
%% Initialize video
myVideo = VideoWriter('Planetary_motion_Asteroid_TP_2'); % open video file
myVideo.FrameRate = 3;
open(myVideo)
% NOTA: - No represento la trayectoria del sol con respecto a SSB porque es
% despreciable en comparaciÛn al resto y adem·s se ve eclipsado por la
% representaciÛn magnificada.
for i=1:5:length(r_Bepi_SSB_eph(:,1)) % Animation accelerated 5 times the normal step (celestial
bodies moving at 9000 min/step = 6.25 days/step)
drawnow;
plot3(r_Bepi_SSB_eph(1:i,1),r_Bepi_SSB_eph(1:i,2),r_Bepi_SSB_eph(1:i,3))
hold on
% plot3(r_Ast_SSB_2BP(1:i,1),r_Ast_SSB_2BP(1:i,2),r_Ast_SSB_2BP(1:i,3))
set(gca,'Color','k')
[X,Y,Z] = sphere;
Sun =
surf(X*R_S+r_Sun_SSB_eph(end,1),Y*R_S+r_Sun_SSB_eph(end,2),Z*R_S+r_Sun_SSB_eph(end,3),'edgealpha',0);
set(Sun, 'FaceColor', [1 1 0])
   % MERCURY
   if floor(i/TP_Mercury) == index_Mer
plot3(r_Mercury_SSB_eph(k_Mer:i,1),r_Mercury_SSB_eph(k_Mer:i,2),r_Mercury_SSB_eph(k_Mer:i,3),'w','Lin
eWidth',1.5)
      aux_Mer = 1;
       j_Mer = i;
   else
       index_Mer = index_Mer+2*aux_Mer;

plot3(r_Mercury_SSB_eph(j_Mer:i,1),r_Mercury_SSB_eph(j_Mer:i,2),r_Mercury_SSB_eph(j_Mer:i,3),'b','Lin
eWidth',2.5)
       aux_Mer = 0;
       k_Mer = i;
   end
   % VENUS

   if floor(i/TP_Venus) == index_Ven
plot3(r_Venus_SSB_eph(k_Ven:i,1),r_Venus_SSB_eph(k_Ven:i,2),r_Venus_SSB_eph(k_Ven:i,3),'w','LineWidth
',1.5)
      aux_Ven = 1;
       j_Ven = i;
   else
       index_Ven = index_Ven+2*aux_Ven;
plot3(r_Venus_SSB_eph(j_Ven:i,1),r_Venus_SSB_eph(j_Ven:i,2),r_Venus_SSB_eph(j_Ven:i,3),'b','LineWidth
',2.5)
       aux_Ven = 0;
       k_Ven = i;
   end
   % Earth
   if floor(i/TP_Earth) == index_Earth
plot3(r_Earth_SSB_eph(k_Earth:i,1),r_Earth_SSB_eph(k_Earth:i,2),r_Earth_SSB_eph(k_Earth:i,3),'w','Lin
eWidth',1.5)
      aux_Earth = 1;
       j_Earth = i;
   else
       index_Earth = index_Earth+2*aux_Earth;
plot3(r_Earth_SSB_eph(j_Earth:i,1),r_Earth_SSB_eph(j_Earth:i,2),r_Earth_SSB_eph(j_Earth:i,3),'b','Lin
eWidth',2.5)
       aux_Earth = 0;
       k_Earth = i;
   end
   % Mars
   if floor(i/TP_Mars) == index_Mars
plot3(r_Mars_SSB_eph(k_Mars:i,1),r_Mars_SSB_eph(k_Mars:i,2),r_Mars_SSB_eph(k_Mars:i,3),'w','LineWidth
',1.5)
      aux_Mars = 1;
       j_Mars = i;
   else
       index_Mars = index_Mars+2*aux_Mars;
plot3(r_Mars_SSB_eph(j_Mars:i,1),r_Mars_SSB_eph(j_Mars:i,2),r_Mars_SSB_eph(j_Mars:i,3),'b','LineWidth
',2.5)
       aux_Mars = 0;
       k_Mars = i;
   end
axis equal
xlabel('X (km)','FontSize',18)
ylabel('Y (km)','FontSize',18)
zlabel('Z (km)','FontSize',18)
title('BepiColombo spacecraft trajectory from October 20th of 2018 to November 2nd of
2025','FontSize',18)
frame = getframe(gcf); %get frame
```

```
    writeVideo(myVideo, frame);

end
close(myVideo)
end
end
```

distances_BepiColombo.m

```
function [time_within_PZ,day_start,day_finish,path_sections,times] =
distances_BepiColombo(r_Mercury_SSB_eph,r_Venus_SSB_eph,r_Earth_SSB_eph,r_Mars_SSB_eph...
    ,r_Jupiter_SSB_eph,r_Saturn_SSB_eph,Propagation_time,r_Bepi_SSB_eph)

distances = zeros(6,length(r_Bepi_SSB_eph));
times = 0:6:(Propagation_time-1)*6;
thresholds = [6.11e6 2.25e7 2.48e7 8.87e6 5.06e8 2.69e8 2.75e6]; % Mercury, Venus, Earth, Mars,
Jupiter, Saturn & Moon (in this order)
for i=1:length(r_Bepi_SSB_eph) % [days]
distances(1,i) = norm(r_Bepi_SSB_eph(i,:)-r_Mercury_SSB_eph(i,:)); % Mercury
distances(2,i) = norm(r_Bepi_SSB_eph(i,:)-r_Venus_SSB_eph(i,:));   % Venus
distances(3,i) = norm(r_Bepi_SSB_eph(i,:)-r_Earth_SSB_eph(i,:));   % Earth
distances(4,i) = norm(r_Bepi_SSB_eph(i,:)-r_Mars_SSB_eph(i,:));    % Mars
distances(5,i) = norm(r_Bepi_SSB_eph(i,:)-r_Jupiter_SSB_eph(i,:)); % Jupiter
distances(6,i) = norm(r_Bepi_SSB_eph(i,:)-r_Saturn_SSB_eph(i,:));  % Saturn
end
for i=1:6
    figure
    hold on
    plot(times(1,:),distances(i,:))
    if i==1
    plot(times(1,:),ones(1,length(r_Bepi_SSB_eph))*thresholds(i));
    title('\fontsize{24}Instantaneous distances from BepiColombo to Mercury along its trajectory')
    elseif i==2
    plot(times(1,:),ones(1,length(r_Bepi_SSB_eph))*thresholds(i));
    title('\fontsize{24}Instantaneous distances from BepiColombo to Venus along its trajectory')
    elseif i==3
    plot(times(1,:),ones(1,length(r_Bepi_SSB_eph))*thresholds(i));
    title('\fontsize{24}Instantaneous distances from BepiColombo to Earth along its trajectory')
    elseif i==4
    plot(times(1,:),ones(1,length(r_Bepi_SSB_eph))*thresholds(i));
    title('\fontsize{24}Instantaneous distances from BepiColombo to Mars along its trajectory')
    elseif i==5
    plot(times(1,:),ones(1,length(r_Bepi_SSB_eph))*thresholds(i));
    title('\fontsize{24}Instantaneous distances from BepiColombo to Jupiter along its trajectory')
    elseif i==6
    plot(times(1,:),ones(1,length(r_Bepi_SSB_eph))*thresholds(i));
    title('\fontsize{24}Instantaneous distances from BepiColombo to Saturn along its trajectory')
    end
    xlabel('\fontsize{20}Time (day)')
    ylabel('\fontsize{20}Distances (km)')
end
%% Loop for accounting times that S/C spends inside of each Perturbative Spheres
time_within_PZ = zeros(1,6); % Mercury, Venus, Earth, Mars, Jupiter & Saturn
day_start = zeros(7);        % Mercury, Venus, Earth, Mars, Jupiter & Saturn (each row is one planet
and each column is a inferiorly crossing)
day_finish = zeros(7);       % Mercury, Venus, Earth, Mars, Jupiter & Saturn (each row is one planet
and each column is a upperly crossing)
for i=1:6
    k = 1;
    aux = 1;
    for j=1:length(r_Bepi_SSB_eph)
        index = distances(i,j);
        if index <= thresholds(i) % When threshold is inferiorly exceeded, it adds one step (one day)
            time_within_PZ(1,i) = time_within_PZ(1,i)+1;
            if k == 1 % Conditional nest to store the day where the threshold is initally inferiorly
exceeded
               day_start(i,aux) = j;
               k = 0;
            end
            if j == length(r_Bepi_SSB_eph)
                break
            end
            if distances(i,j+1) > thresholds(i) % Conditional nest to store the day where the
threshold is initally upperly exceeded
               day_finish(i,aux) = j+1;
               k = 1;
               aux = aux+1;
            end
        end
        if j == length(r_Bepi_SSB_eph)
                break
        end
    end
end
%% Loop for accounting distances covered by the S/C inside of a Perturbative Sphere (in real, it
computes the position vector changes)
% Loop for substituting zero values by ones (in order to have an allowed
% index):
for i=1:length(day_start)
    for j=1:length(day_finish)
```

```
        if day_start(i,j) == 0
            day_start(i,j) = 1;
        end
        if day_finish(i,j) == 0
            day_finish(i,j) = 1;
        end
    end
    if i == 1 % For allowing an extra path section in the case of Mercury (arrival)
        day_finish(i,j) = length(r_Bepi_SSB_eph);
    end
end
path_sections = zeros(7); % Mercury, Venus, Earth, Mars, Jupiter & Saturn (each row is one planet and
each column is a path section)
% NOTE: - Each path section is the section of the trajectory where the S/C
% is strictly inside a perturbative zone (sphere)
for i=1:6
    for j=1:length(day_start)
        if day_start(i,j) < day_finish(i,j)
           path_sections(i,j) = norm(r_Bepi_SSB_eph(day_finish(i,j),:)-
r_Bepi_SSB_eph(day_start(i,j),:));
        end
    end
end
end
```

gravitational_fields_BepiColombo.m

```
function gravitational_fields_BepiColombo(G,r_Bepi_SSB_eph,r_Sun_SSB_eph,r_Mercury_SSB_eph...
    ,r_Venus_SSB_eph,r_Earth_SSB_eph,r_Mars_SSB_eph,r_Jupiter_SSB_eph,r_Saturn_SSB_eph...
    ,Mass,Propagation_time)
% We apply the N-body problem in order to compute the asteroid-to-Sun
% instantaneous relative acceleration, taking into account the planets as
% perturbative bodies (so we do not need to integrate it).
% Key subindex:
% 1:Sun 2:BepiColombo spacecraft 3:Mercury 4:Venus 5:Earth 6:Mars 7:Jupiter
% 8:Saturn 9:Uranus 10:Neptune 11:Pluto (8 - 11 not yet included)
B_vector = zeros(1,length(r_Bepi_SSB_eph)); % Value of perturbation term for BepiColombo's positions
times = 0:6:(Propagation_time-1)*24;
threshold = 6.94e-10; % Threshold computed for BepiColombo [km/s2]
for i=1:length(r_Bepi_SSB_eph)
    % First update each position vector:
    r31 = r_Sun_SSB_eph(i,:)-r_Mercury_SSB_eph(i,:); % Sun relative to Mercury
    r41 = r_Sun_SSB_eph(i,:)-r_Venus_SSB_eph(i,:);   % Sun relative to Venus
    r51 = r_Sun_SSB_eph(i,:)-r_Earth_SSB_eph(i,:);   % Sun relative to Earth
    r61 = r_Sun_SSB_eph(i,:)-r_Mars_SSB_eph(i,:);    % Sun relative to Mars
    r71 = r_Sun_SSB_eph(i,:)-r_Jupiter_SSB_eph(i,:); % Sun relative to Jupiter
    r81 = r_Sun_SSB_eph(i,:)-r_Saturn_SSB_eph(i,:);  % Sun relative to Saturn
    r12 = r_Bepi_SSB_eph(i,:)-r_Sun_SSB_eph(i,:);    % BepiColombo relative to Sun
    r32 = r12+r31;                                   % BepiColombo relative to Mercury
    r42 = r12+r41;                                   % BepiColombo relative to Venus
    r52 = r12+r51;                                   % BepiColombo relative to Earth
    r62 = r12+r61;                                   % BepiColombo relative to Mars
    r72 = r12+r71;                                   % BepiColombo relative to Jupiter
    r82 = r12+r81;                                   % BepiColombo relative to Saturn
    % Computing indirect gravitational term (B):
    B = -G*Mass(1)*(r32/norm(r32)^3-r31/norm(r31)^3)-G*Mass(2)*(r42/norm(r42)^3-r41/norm(r41)^3)...
        -G*Mass(3)*(r52/norm(r52)^3-r51/norm(r51)^3)-G*Mass(4)*(r62/norm(r62)^3-r61/norm(r61)^3)...
        -G*Mass(5)*(r72/norm(r72)^3-r71/norm(r71)^3)-G*Mass(6)*(r82/norm(r82)^3-r81/norm(r81)^3);
    % Since we are no longer interested in the vector direction but in its
    % magnitude:
    B_vector(i) = norm(B);
end
figure
hold on
plot(times,B_vector,'-b','LineWidth',2)
plot(times(1,:),ones(1,length(r_Bepi_SSB_eph))*threshold)
xlabel('\fontsize{20}Time (day)')
ylabel('\fontsize{20}Perturbing acceleration term (B) [km/s^2]')
xlim([0 length(times)*6])
% ylim([1e-11 30e-10]) % Limit to properly see the crossings on the threshold
title('\fontsize{24}Magnitude of instantaneous perturbative acceleration exerted by planets in the
Spacecraft')
end
```